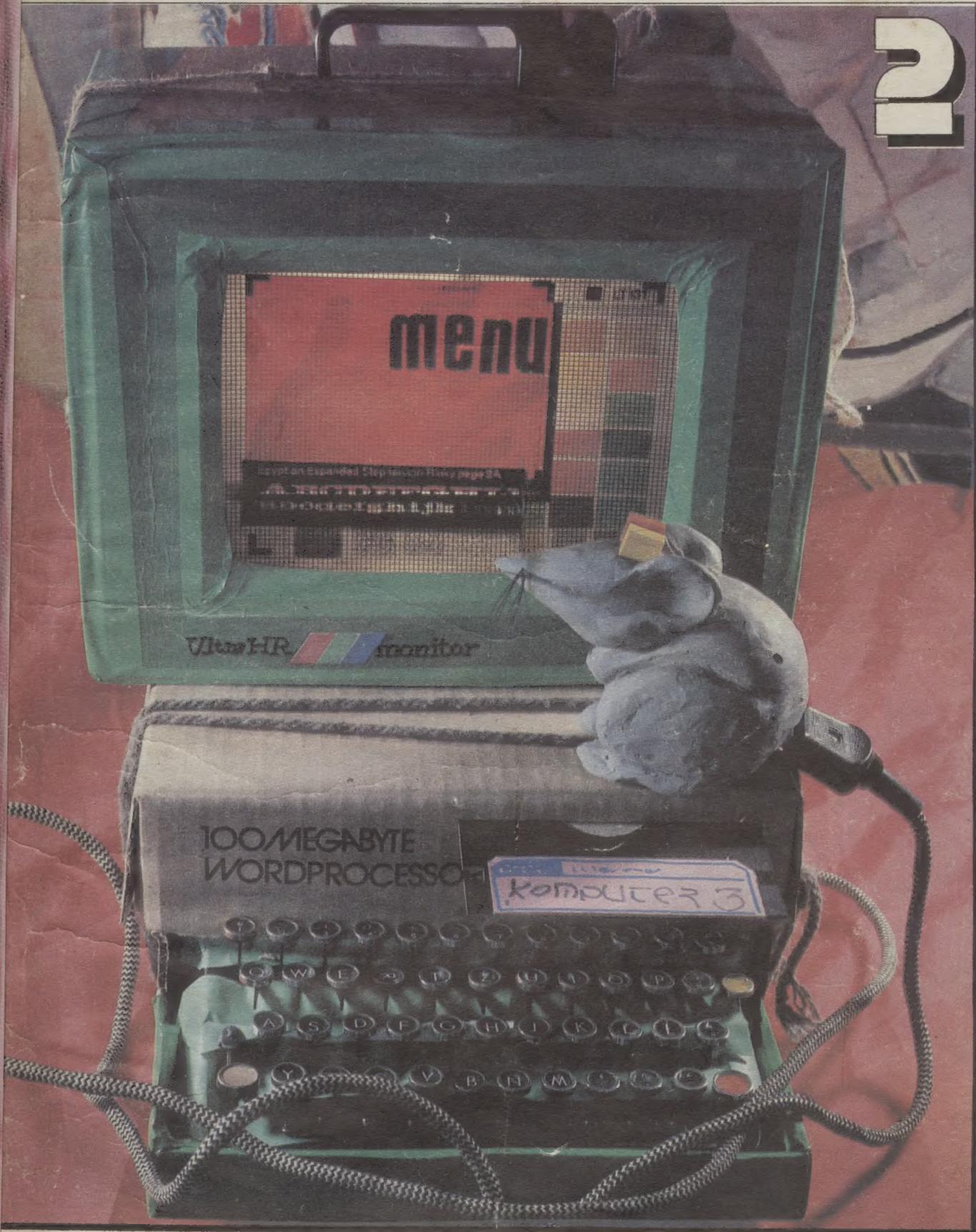


# компьютер

ISSN 0236-1949

2



# Компьютер

Финансы и статистика и **компьютер**

Москва, 1990

УДК 681.3 (082)

Сборник статей

Издается с 1990 г.

Выходит ежеквартально

Выпуск 2

Ответственные за выпуск:

К.Коробов, М.Цар

Редакторы: О.Ермилина, А.Поплавский

Художественный редактор: С.Щипка

Программист: М.Сэтляк

Корректор: Т.Васильева

Адрес редакции:

101000 Москва, ул. Чернышевского 7.

Издательство "Финансы и статистика",  
редакция сборника "Компьютер"

Телефоны: 209-99-94; 925-35-02

Набор:

**NEXT** sp. z o.o., Варшава,

ул. Сенаторска 40/46, тел. 27 12 85

Настольная издательская система:

Atari Mega ST2, лазерный принтер,  
сканер CANON IX 12F,

программа: DMC № 080247

*calamus*

Подписано в печать 29.05.90 г.

Формат 60\*90 1/8. Бумага офсетная № 1.

Гарнитура "Seti 70r". Печать офсетная.

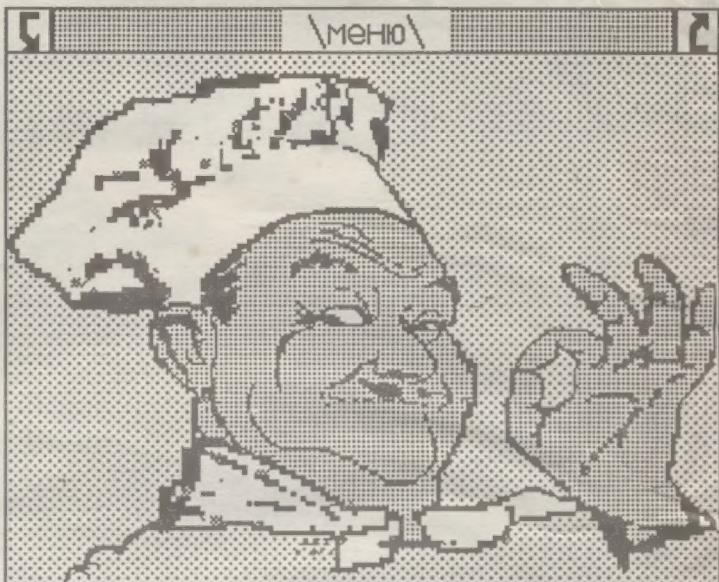
Усл.-печ. л. 8,0. Усл.-кр. отт. 26,0. Уч.-изд.  
л. 11,52. Тираж 100.000 экз. Заказ №

Цена 1 руб. 50 к.

Чеховский полиграфический комбинат  
142300 г. Чехов Московской обл., ул.  
Полиграфистов, д. 1

ВОЗВРАТИТЕ КНИГУ НЕ ПОЗЖЕ  
обозначенного здесь срока

23.02.91-928		27-25.07.93	
14.01.82		20.2.86-12.6	
96-18.09.93		23.08.93-3	
27-8.03.95			



## В мире "Компьютера"

### NOVELL ставит сеть

Марек Цар

5

### Стивен Джобс

Халина Мадейчик

7

### Как я попался в сеть

Якуб Татаркевич

8

### Технология баз данных

Михаил Когаловский

10

### Новинки из долинки

Ежи Оркишевский

15

### ELBlkey, File-PROTECTION

С. Груздев, А. Сучков

17

### ИРИС

С. Бебчук, В. Соркин

19

## Компьютер на работе

Мониторы ...

Юрий Кобленц-Мишке

21

... и адAPTERы

Михаил Сальников

24

Карта телефонного  
модема

Яцек Гракх

28

Сети NOVELL

Казимеж Цянцияра

87457  
29

ТЕХНИЧЕСКИЕ

ВЫВОДЫ

Петрозаводское

Октябрьское

АЛИСА+

В. Балинский, Е. Кривопальцев

31

Программы-упаковщики

Ян Стожек

33

Компьютерный склад

информации

36

М.Дец, М.Матушак, М.Млынарски

Выбор принтеров

Юлия Сальникова

38

Принтер STAR LC-15

Зенон Рудак

40

На книжную полку

42

## Пишем файл CONFIG.SYS

Виктор Фигурнов

43

## Norton Commander 3.0

Ян Рожанович

45

Антивирусные  
программы

Марек Сель

48

## Компьютер дома

## Коротко о FidoNet

Ян Стожек

51

Палка, приносящая  
радость

Анджей Поплавский

53

Немного об экспертных  
системах

55

Кириллица на  
Спектруме

Владислав Крамаренко

57

## Сказка о Драконе

Лешек Рудак

59

## Секреты Atari XL/XE

Петр Грабчиньский

63

## Что, когда, где

### АВГУСТ

#### СОМТЕС'90

Международная выставка компьютерной техники

23-26 августа

Сингапур

Организатор:

ITP - Services PTE Ltd, 994  
Bendemeer Road, Suite 04-03,  
Singapore 1233

телефон 291-32-38

телефон 55223

факс 2965384

**СОВЕЩАНИЕ** "Автоматизация информационно-библиотечной технологии и экономика библиотечного дела"

август, 3 дня

Кишинев, СССР

Организатор:

Центральная научная библиотека АН Молдавской ССР, 277012, г. Кишинев, 12, пр-т Ленина, 1  
телефон 26-27-91

### СЕНТЯБРЬ

#### PCW'90

Personal Computer World Show

6 дней

Лондон, Великобритания

Организатор:

Montibuild Exhibitions Ltd, 11  
Manchester Square, LONDON  
W1M 5AB

телефон 01/4861951

телефон 24591

факс 01/4873260

#### SOFTTARG'90

IV Всепольская ярмарка программного обеспечения

18-21 сентября

Катовице, Польша

Организатор:

Osródek Postępu Technicznego, 40-147 Katowice, ul. Buczka 1b

телефоны 598312, 596061  
доб. 264

телефон 0312458 opt pl

факс 588919

#### SYSTEMOTECHNIKA'90

6 Международная выставка систем для rationalизации производства и управления, интеграции ЭВМ и информационной техники с сопровождающей конференцией по теме "менеджмент"

17-22 сентября

Москва, СССР

Организатор:

Glahe International KG, Herler  
Strasse 103-109, D-5000  
Koeln 80, BRD

телефон 0221/694011

телефон 8 874 676

факс 0221/695865

**СЕМИНАР** "Создание интеллектуальных САПР, СВИС и электронных средств"

сентябрь, 3 дня

г. Геленджик, СССР

Организатор:

ЦП ВНТО радиотехники, электроники и связи им. А.С. Попова, 103897  
Москва, ГСП-3, ул. Кузнецкий мост, 20

телефоны 921-71-08,  
924-70-72

**СОВЕЩАНИЕ** "Проблемы информатики и вычислительной техники в строительстве"

сентябрь, 3 дня

Москва, СССР

Организатор:

ЦП ВНТО строительной индустрии, 103062 Москва, Подсосенский пер., 25  
телефон 297-97-89

**КОНФЕРЕНЦИЯ** "Практическое применение современных технологий программирования, пакетов прикладных программ в вычислительных системах и сетях ЭВМ"

сентябрь, 3 дня

Днепропетровск, СССР

Организатор:

НПО "Орбита", 320700, Днепропетровск, 126, ул. М. Паниахи, 2

телефон 65-78-69

### ОКТЯБРЬ

#### ORGATECHNIK'90

Международная ярмарка оргтехники

25-30 октября

Кельн, ФРГ

Организатор:

Messe- und Ausstellungsgesellschaft mbH, 5000 Koeln 21, Messeplatz 1, Postfach 210760

телефон 0221/8210

телефон 8873426

факс 0221/8212574

#### SCANCOM'90

Computer & Telecommunications Show

23-26 октября

Готенбург, Швеция

Организатор:

Svenska Mässan, The Swedish Exhibition Centre, Gothenburg, P.O. Box 5222, S-402 24 Gothenburg, Швеция

#### SYSTEC'90

3 Международная ярмарка компьютерной интеграции в промышленности

22-26 октября

Мюнхен, ФРГ

Организатор:

Muenchener Messe- und Ausstellungsgesellschaft mbH, 8000 Muenchen 12, Postfach 121009

телефон 089/5107-0

телефон 5212086 ameg d

факс 089/5107-506

#### INFORMACJA'90

II Международная компьютерная ярмарка

10-12 октября

Катовице, Польша

Организатор:

Przedsiębiorstwo Techniczno-Uslugowo-Promocyjne PRO-INFO, 40-530 Katowice ul. Drozdow 31

телефон 832/534 288

телефон 312401 info pl

факс 832/519-634

## ПРОМЫШЛЕННАЯ ЯРМАРКА

23-26 октября

Киев, ВДНХ

Организатор:

Przedsiębiorstwo Techniczno-Uslugowo-Promocyjne PRO-INFO, 40-530 Katowice ul. Drozdow 31, Польша

телефон 832/534 288

телефон 312401 info pl

факс 832/519-634

## ЛОКСЕТЬ '90

IV Конференция "Локальные вычислительные сети"

октябрь, 3 дня

Юрмала, Латвия

Организатор:

Институт электроники и вычислительной техники АН Латвийской ССР, 226006 Рига, ул. Академия, 14

телефоны 52-89-43,  
52-28-28

## II КОНФЕРЕНЦИЯ ПО ИСКУССТВЕННОМУ ИНТЕЛЛЕКТУ

октябрь-ноябрь, 3 дня

Минск, БССР

Организатор:

Институт технической кибернетики АН БССР, 220605 Минск, ул. Типографская, 6

телефон 39-51-75

## СЕМИНАР Программное обеспечение СМ ЭВМ

октябрь, 3 дня

Москва, ВДНХ СССР

Организатор:

НПО "Электронмаш", 117812 Москва, ул. Вавилова, 24

телефон 939-00-54

- NOVELL ставит сеть
- Стивен Джобс
- Как я попался в сеть
- Технология баз данных
- Новинки из долинки
- Компьютеризируемся

В мире "Компьютера"

## NOVELL ставит сеть

С Эндрю Зольтовски, директором NOVELL UK Ltd. беседует (не только о сети NetWare) Marek Цар

М.Ц. Хотя, как мне кажется, искушенных читателей "Компьютера" знакомить с изготовителями самой популярной в мире сетевой операционной системы Advanced NetWare 286 не надо, все-таки в самом начале нашей беседы я попрошу Вас вкратце рассказать о фирме NOVELL.

Э.З. С 1985 г. наша фирма специализируется на технологии локальных сетей. Мы первыми решили применять на микрокомпьютерах концепцию работы file server (узел локальной сети, выполняющий определенные функции по запросам других узлов). Раньше на этом уровне компьютерной техники пользователи одного жесткого диска вынуждены были делиться им в самом прямом смысле - и по времени, и по объему. Разработанная нами сетевая операционная система обеспечивает одновременный доступ нескольких пользователей к программам или файлам данных, предназначенным для коллективного доступа и хранящимся на жестком диске файл-сервера. За нами последовали другие, но NOVELL успела уже захватить 90% мирового рынка сетевых операционных систем. Согласно данным начала 1990 г. в мире работает полмиллиона наших сетей, обслуживающих 5 млн. пользователей. Продукты, выпускаемые фирмой NOVELL, продаются во всем мире, в том числе и в странах Восточной Европы.

М.Ц. А в Советском Союзе?

Э.З. В 1989 г. мы решили, что это один из самых перспективных рынков сбыта наших продуктов. Советским пользователям предлагаются все версии наших операционных систем, в том числе новейшие: NetWare 386, Advanced Netware v 2.15 и ее упрощенный вариант для начинающих - ELS (Entry Level Solution) NetWare Level II v. 2.15.

М.Ц. NOVELL обычно не продает своих изделий конечным пользователям. Этим вопросом занимают-

ся фирмы, представляющие ваши интересы. Придерживаетесь ли вы этого принципа и на советском рынке?

Э.З. Поставщиками наших систем в СССР являются пока четыре фирмы: две английские - RANK XEROX и QUEST AUTOMATION Ltd, MICRONET из Финляндии и BGS из Австрии.

М.Ц. Не препятствует ли вашему выходу на советский рынок американское правительство, которое, как известно, внимательно следит за поставками компьютерной технологии в СССР?

Э.З. Каждая сделка по поставке нашего сетевого программного обеспечения в Советский Союз сопровождается экспортной лицензией, аprobированной



американскими правительственными органами. Я уверен, что без этого предварительного контроля мы могли бы продавать в 10 раз больше, чем сейчас.

М.Ц. Значит, получить экспортную лицензию не так уж легко. Как долго длится оформление соответствующих документов?

Э.З. По-разному, в зависимости от покупающего и того, что именно он хочет купить. Тут нет никаких правил. Иногда с момента отправления соответствующих документов в США до получения нами лицензии проходят недели. Но были случаи, когда ответ приходилось ждать и полгода.

М.Ц. По-моему, такое положение только усугубляет проблему нелегального копирования ваших программных продуктов. Намерены ли Вы, и каким именно образом, бороться с этим?

Э.З. Пиратские копии - вопрос очень эмоциональный. С этим явлением мы имеем дело в первую очередь на "молодых" рынках сбыта программного обеспечения. В Соединенных Штатах самый разгар страсти, связанных с этим явлением, бушевал, примерно, 10 лет назад, в Великобритании - 5 лет назад. Сегодня и мы, и другие фирмы, выпускающие программное обеспечение, отказались от защиты наших продуктов от копирования.

М.Ц. Почему?

Э.З. Поскольку такие меры невыгодны легальным, крупным пользователям. Именно такое требование поставил перед нами американский рынок, где проходит более 60% наших сетей. Мы внимательно следим за положением в этой области в странах Восточной Европы и пришли к выводу, что, например, в Польше пользователи уже начинают, так сказать, "созревать" в этом отношении.

М.Ц. А как бороться с "несозревшими"?

Э.З. Как с любым другим воровством. Но тут нуж-

но учесть еще одно обстоятельство. Иногда пользователи просто не знают о том, что у них пиратская, нелегальная копия. Итак, с проблемой нелегального копирования мы еще имеем дело, но не стоит преувеличивать значение этого вопроса. Сейчас стало ясно, что великолепным средством борьбы с нелегальными копиями стали ... компьютерные вирусы, распространяемые чаще всего с бесплатным (читай "ворованным") программным обеспечением.

М.Ц. Но с копированием программного обеспечения, и в первую очередь столь сложного как ваше, можно бороться созидательным путем. Именно такую цель преследуют центры по обучению работе с вашими программами. Три таких центра уже работают в Польше. А в СССР?

Э.З. Первый советский центр по обучению нашим сетевым системам был организован в конце 1989 г. фирмой QUEST AUTOMATION Ltd. Я надеюсь, что со временем появятся и другие. Мы отлично понимаем, что пользователю не безразлично, какая (легальная или нелегальная) версия нашей программы у него. Ведь только в том случае, если он располагает легальной версией, он сможет рассчитывать на помощь в установке сети, обучение обслуживающего ее персонала, поставку новых версий. Если поставщик обеспечивает весь комплекс перечисленных мною услуг, вопрос нелегального копирования полностью снимается.

М.Ц. Первым шагом в указанном Вами направлении должна быть подготовка технической документации на русском языке.

Э.З. По западным масштабам NOVELL - это относительно небольшая фирма. Во всех ее филиалах, расположенных в Канаде, Великобритании, ФРГ, Франции и Гонконге, занято всего 1500 человек. Мы сосредоточиваем наше внимание на рынках, где сети NetWare пользуются самым большим спросом. Конечно, потенциальные возможности сбыта наших систем на советском рынке очень заманчивы. Однако в Советском Союзе есть в этом отношении и некоторые другие возможности, о которых я не хотел бы говорить более подробно. Если их удастся осуществить - наши пользователи получат документацию на русском языке. Но это вопрос хотя и не столь отдаленного, но все же будущего.

М.Ц. И последний вопрос. 1992 г. - год объединения Западной Европы - не так уж далек. Эта дата будет переломной не только для политиков. Многие считают, что именно в 1992 г. в Европе начнется эра компьютерных сетей. Но во всей ли Европе?

Э.З. О том, что начинается эра компьютерных сетей, я слышу каждый год, начиная с 1982 г. Мне кажется, что в США она началась, примерно с 1984 г., в 1985 г. - в Великобритании, чуть позже в других странах Западной Европы. Это, конечно, весьма субъективное мнение, но оно подкреплено цифрами, отображающими скачкообразный рост продажи сетевых систем. В странах Восточной Европы из-за искусственных препятствий в распространении компьютерной технологии начало этой эры еще впереди. Были и до сих пор еще существуют некоторые рыночные барьеры в самих этих странах. Но все преграды постепенно исчезают. По-моему, еще не все осознали, что сети, и особенно недорогих микрокомпьютеров, - важный инструмент в поддержку перестройки. Ведь именно они подкрепляют процесс демократизации предприятий. 1992 г. может стать началом эры сетей, но в Восточной Европе.

М.Ц. Благодарю Вас за беседу.

\ В мире "Компьютера" \

© Халина Мадейчик

# Стивен Джобс

Мы не станем называть его отцом микроЭВМ. Этим приемом пользуются все журналисты, которые пишут о владельце двух крупных американских компьютерных фирм - APPLE COMPUTER и NEXT. Не стоит также писать о том, что свою первую машину он построил вместе со Стефаном Возняком в гараже в 1977 г. Об этом тоже знают все.

Профессиональная карьера С. Джобса началась в момент основания фирмы APPLE. Но тут же появились и первые проблемы. Ведь точно такое же название и очень похожий фирменный знак были и у фирмы рок-группы "Битлз". Несмотря на судебные разбирательства, которые стали неотъемлемой частью жизни Джобса, ему сопутствовала удача. Популярность разработанных APPLE COMPUTER Inc. микрокомпьютеров Apple, Macintosh, Lisa с каждым днем росла, прежде всего в студенческой среде и среди научного персонала американских ВУЗов. Со временем фирме потребовался менеджер-профессионал. Себя Джобс всегда считал прежде всего "генератором" идей. Управление бизнесом следовало поручить другим.

В 1983 г. Джобс решил привлечь к работе в APPLE COMPUTER Джона Скулея из концерна PEPSICO. Судя по тому, как развернулись дальнейшие события, это было роковое решение. Не помогли задушевные беседы, которые оба они вели во время столь любимых Джобсом прогулок по лесным тропинкам. Первым в 1984 г. покинул APPLE Возняк, заявив, что его "компания потеряла душу". Через два года ушел и сам Джобс. 12 сентября 1985 г. было принято решение об образовании новой

фирмы - NEXT Inc.

APPLE COMPUTER потеряла не только душу. Вместе с Джобсом из фирмы ушли еще четверо его сотрудников. "Меня тут же обвинили в воровстве идей и технологий, - вспоминает Джобс. - Причем кто?! Один из самых близких моих друзей, основоположник APPLE COMPUTER Майк Маркьюла. На самом же деле мы просто не знали, с чего будем начинать".

А начинать можно было уже со значительно более высокого уровня, чем в 1977 г. Тогда, чтобы купить микросхемы для первого микрокомпьютера, Джобс был вынужден продать свой старый "Фольксваген". Основной же капитал NEXT Inc. составлял 100 млн. дол.

Если "Яблочко" строилось в гараже, то планы нового компьютера - NeXT (New XT, как его иногда расшифровывают) - вынашивались в шикарной резиденции Джобса в горах Пало-Алто в Калифорнии. Кроме Джобса (67%), в NEXT свои капиталы вложили мультимиллиардер из Техаса Генри Росс Перот (16%), японская фирма CANON Inc., разработавшая для NeXT лазерный принтер и дисководы оптических дисков (16%), а также весьма известные американские фирмы GENERAL MOTORS и ELECTRONIC DATA SYSTEMS (EDS) и даже два университета - Станфорд и Карнеги-Меллон (правда, только по 1%).

Прошло три года и как всегда элегантный и одетый по последней моде Джобс заявил всему миру о своем возвращении. В октябре 1988 г. состоялась пресс-премьера его новой машины. Будет ли она столь же "удачливой", как и джобсовское "яблочко" - покажет время.



# Как я попался в сеть

В Институте им. Макса Планка (г. Штуттгарт), где я работал больше года, мне довелось познакомиться с "раем" для пользователей компьютеров. Я имею в виду пользователей из Польши, ибо для большинства научных сотрудников, работавших со мной в институте, компьютер - главное орудие труда. Под словом "компьютер" я подразумеваю не только машину типа IBM 370, а прежде всего находящийся в Мюнхене суперкомпьютер Cray X-MP. И хотя формально гражданам из стран Восточной Европы нельзя пользоваться суперкомпьютерами, однако на практике доступ к ним возможен (правда, под контролем штатного сотрудника института).

Связаться с мюнхенским суперкомпьютером очень просто. На каждом этаже института есть так называемая компьютерная комната, в которой стоят терминалы типа DEC VT 220. Кроме того, такие терминалы имеются во многих лабораториях, а те, кто занимается графикой, могут воспользоваться графическими терминалами типа Hirz. Новый пользователь должен записаться в "компьютерной канцелярии" EDV (Elektronische Daten Verarbeiten - электронная обработка данных) и получить свою "метку" - чаще всего сокращение фамилии или имени. Я пользовался меткой "Куба" - уменьшительная форма имени Якуб, так как моя фамилия оказалась слишком длинной... Кроме этого, каждый, кто хочет работать в локальной сети, должен (для безопасности) иметь свой пароль. Его не знают даже операторы системы (хотя могли бы узнать, заглянув внутрь системы, но это не принято...). Сейчас я могу уже сказать, что моим паролем было слово *Rudy* - оно легко набиралось на клавиатуре и в то же время никак не ассоциировалось со мной (*Rudy* по-польски значит "рыжий", немцы же воспринимают это слово как имя. - Примеч. пер.).

Как выглядит контакт с компьютером в сети? Как правило, терминалы работают круглые сутки, поскольку электроника "не любит", когда ее часто включают и выключают. Однако я не советовал бы оставлять оборудование постоянно включенным там, где часто исчезает напряжение в сети - для наших драгоценных игрушек это может быть убийственно. Итак, когда я приходил в компьютерную комнату, экраны терминалов обычно были погашены автоматически - выключателями пучка электронов, установленными для того, чтобы избежать выжигания статического изображения и продлить "жизнь" кинескопов. Для начала мне надо было нажать клавишу "BBO". Появлялся экран с информацией об ОС и просьбой ввести метку и пароль. Когда набирается пароль, курсор перемещается, но буквы при этом не видны. Таким образом, постороннее лицо не может узнать ваш пароль. Если метка или пароль написаны неправильно, компьютер вежливо сообщает, что такого пользователя в списке нет, и просит повторить ввод. Если все правильно, то на дисплее появляется приветствие, последние системные сообщения от операторов, а

также информация о запланированных выключениях компьютера и о компьютерной почте, поступившей в адрес пользователя (но об этом позже).

Работу начинаем, например, с вызова текстового редактора, а потом одного из компиляторов (Фортран, Паскаль, TeX). Таким образом настраиваем суперкомпьютер на выполнение определенной задачи или на распечатку на лазерном принтере. К некоторым терминалам присоединены обычные матричные принтеры, на которых можно получить черновую копию работы.

Однако настоящее удовольствие от работы получаешь, используя возможность выхода на европейскую сеть научной информации EARN, а через нее - на самую популярную в научном мире сеть BITNET. Далее можно выйти на сеть Департамента обороны США (ARPANET) и другие сети. Полная свобода контактов почти со всем миром (почти, поскольку с Польшей, например, контакта нет...). Остается надеяться, что и это изменится.

Отправление компьютерного письма в этой системе чрезвычайно просто. Надо приготовить адресный список - для этого есть специальная программа, дающая вопросы после введения слова NAMES. Типичные элементы этого списка таковы:

1) псевдоним, который будет использоваться впоследствии при отправлении писем (в моем списке были Янек, Крис, Збышек, Ричард, Петер и т.д.);

2) метка - иногда короткая, например "Куба", а если адресат работает в обширной компьютерной системе - подлиннее;

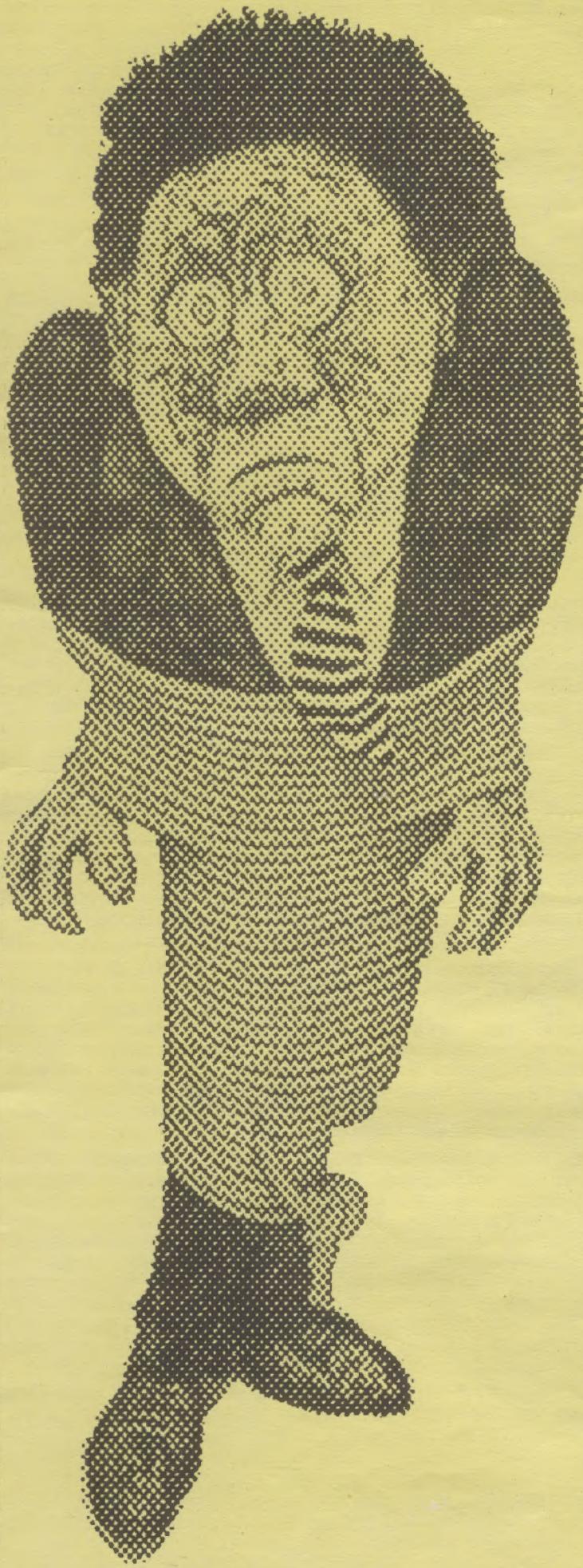
3) название "рукава", в котором работает адресат. Мой "рукав" назывался DSØMPII, что расшифровывается как Германия (D), Штуттгарт (S), Институт им. Макса Планка (MPI), номер лаборатории (II).

В список можно ввести фамилию адресата (это важно не для компьютера, а скорее для людей, получающих письма на один и тот же адрес), номера телефонов (это также несущественно, ибо связаться через компьютер можно быстрее и, прежде всего, дешевле: ведь EARN и BITNET - бесплатные сети). И это отнюдь не благородительность, а вежливость телефонных компаний по отношению к людям науки, благодаря которым существуют компьютеры, сети и т.д.).

Само отправление письма также просто: надо ввести слово MAIL, а затем псевдоним адресата. Операционная система запрашивает тему письма (исключительно для удобства адресата, а не для цензуры...) и выдает пустую страницу программы-редактора. Теперь надо набрать текст и заявить готовность к отправке. Компьютер на всякий случай просит подтвердить, что письмо действительно готово (может случиться, что в последний момент человек вспомнит что-то очень важное), и спустя несколько минут выдаст сообщение, что письмо уже дошло до места назначения. Просто, не правда ли? После недельного пользования такой почтой совершенно не хочется возвращаться к бумаге, конвертам и маркам...

Для того чтобы описываемая картина не показалась вам слишком розовой, добавлю, что иногда, в результате сбоев, письма идут гораздо дольше (например, несколько часов). Но решением этого вопроса в сетях занимаются специальные компьютеры, проверяющие адреса и, по мере необходимости, повторяющие передачу.

В "почтовом режиме" адресату необязательно находиться у терминала. Письмо будет "отложено" в специальную часть жесткого диска. Типичная картинка из "компьютерной комнаты": человек садится к терминалу и принимает целый ряд писем. Некоторые из них распечатывают на принтере, на другие отвечает сразу. Сплошное удовольствие!



Для самых нетерпеливых предусмотрен еще и "разговорный режим": возможность прямого контакта с коллегой, сидящим у терминала в соседнем здании или на другом континенте. Один из моих товарищ любил (самым что ни на есть компьютерным образом) пофлиртовать с симпатичной программисткой из... Калифорнии, хотя обычно такие контакты используются для вполне серьезных целей, например, для обмена информацией о новейших экспериментах. Именно по таким каналам шли самые "свежие" сообщения об открытиях в области "холодного" термоядерного синтеза, которые побуждали людей ускорить работу лучше, чем какие-либо публикации, и быстрее, чем спутниковое телевидение!

И еще один "подарок" сети EARN - базы данных, содержащие общедоступные программы (freeware), программы типа shareware, а также усовершенствования для коммерческих продуктов (нередко помещаемые в базы фирмами-производителями, для которых это и есть кратчайший путь к клиенту). Мне удалось убедить своих коллег, работающих, как и я, на компьютерах Macintosh, подключить их к центральному компьютеру в качестве терминалов. Над этим пришлось немного потрудиться, так как в компьютерной системе института оборудование снабжено необходимыми разъемами RS 232, но работает на сигналах напряжением около 15 В, а машины фирмы APPLE генерируют сигналы напряжением 5 В. Надо было пропустить их через специальный усилитель, который, к счастью, мы быстро получили от фирмы, обслуживающей институтские ПК. И началось ежедневно мы устанавливали связь с базой USTSERV, находящейся в американском университете Райс. Оттуда мы получили, в частности, прекрасные антивирусные программы, благодаря которым удалось избежать "инфекции", занесенной в наши "Маки" одним из студентов, а также выявить вирус на диске, присланной мне из Польши (хотя вирус оказался американским!).

Если уж мы в контексте сети коснулись вирусов, то мне вспоминается такой случай: после того, как в пентагоновской сети ARPANET появился вирус, о котором вскоре заговорила вся мировая печать, операторы опасались, как бы он не забрался и в нашу локальную сеть. Больше месяца без шумихи и лишних разговоров проверялись буквально все приходившие из США компьютерные письма.

Второй случай касался лично меня. В один прекрасный день я получил "по цепочке" письмо от незнакомого человека из Голландии. В нем предлагалось отправить письма такого же содержания еще двадцати адресатам. В случае, если бы мне вдруг взбрело в голову прервать цепочку, автор предсказывал ужасные последствия. Я написал незнакомцу (адрес я узнал из заголовка - компьютер не принимает анонимок...), что его "творчество" может плохо кончиться для всей сети и предложил ему перестать так развлекаться. А кроме этого, я задал вопрос, почему он направил письмо именно мне. Спустя несколько часов пришел ответ с извинениями за глупое поведение. Однако, почему первое письмо попало именно ко мне, я так и не узнал. Ну что ж, может быть, я просто попался в сеть...

Перевод Анджея Поплавского



# Технология баз данных

С самых первых шагов интенсивно развивающееся производства персональных ЭВМ системы управления базами данных стали неотъемлемым компонентом инструментального обеспечения новых информационных технологий, возникших благодаря чрезвычайно богатым возможностям этого замечательного оборудования.

Сегодня рынок программного обеспечения ПЭВМ располагает большим числом разнообразных по своим функциональным возможностям коммерческих систем управления базами данных практически для всех массовых моделей машин и предназначенных для различных операционных систем.

В области "персональных" баз данных сосредоточен в настоящее время мощный потенциал разработчиков программного обеспечения. Помимо таких крупных фирм, как ASHTON-TATE Corp., MICRORIM Inc., BORLAND INTERNATIONAL Inc., NANTUCKET Inc., FOX SOFTWARE Inc., которые вошли в "клуб" производителей СУБД только в 80-е годы, сюда устремились и фирмы, обладающие многолетним опытом разработки таких средств для больших вычислительных систем и мини-ЭВМ. Фирма ORACLE Corp. перенесла в среду ПЭВМ и продолжает развитие своей широко известной системы Oracle [1]. Фирма BM Corp. предоставила свою СУБД DB2 [2] и пользователям ПЭВМ PS/2. Разработанная в Калифорнийском университете для ЭВМ PDP-11 система Ingres перенесена фирмой RELATIONAL TECHNOLOGY Inc. в среду ПЭВМ [3].

Рынок программного обеспечения "персональных" баз данных отличается острой конкурентной борьбой, оперативной реакцией на потребности пользователей и упущениями соперников. Так, имеющая огромное количество пользователей система dBaseII PLUS (ASHTON-TATE Corp.) [4] за годы ее поставки "обросла" большим количеством созданных другими фирмами программных продуктов, практически совместимых с этой системой и в некоторых аспектах ее превосходящих.

Система dBaseII PLUS предоставляет пользователям весьма развитый язык программирования и интерпретатор этого языка. Режим интерпретации, конечно, удобен для разработки и отладки программ, но он приводит к значительной потере производительности на стадии исполнения. Радикальное решение этой проблемы не обеспечивается предусмотренными в этой СУБД средствами псевдокомпиляции.

В этой связи вслед за системой dBaseII PLUS, которая успешно освоила и существенно расширила рынок своих предшественниц dBaseII и dBaseIII, появилась целая группа компиляторов для языков, близких к языку dBaseII PLUS, - Clipper (NANTUCKET Inc.), FoxBase+ (эффективный псевдокомпилятор фирмы FOX SOFTWARE Inc.), Quick Silver (WORD TECH SYSTEM Inc.) и других. Среди них важным достоинством отличается Clip-

per, способный создавать быстрые приложения в виде стандартных для операционной системы загрузочных модулей, полностью "отчуждаемых" от инструментальных средств. Однако этому компилятору свойственен и серьезный недостаток - большой размер загрузочного модуля (более 110 Кбайт) даже при скромном размере исходного кода. Это обстоятельство привело к появлению нового компилятора - dBFast (dBFast Inc.) [5,28], порождающего маленькие загрузочные модули.

Можно было бы продолжать рассмотрение этой "цепной реакции". Но мы ограничимся здесь лишь информацией об ответном шаге ASHTON-TATE Corp., которая выпускает программный продукт STEP IV-ward [6], конвертирующий записанные в языках Clipper, FoxBase+ и QuickSilver приложения в язык своей новой СУБ дBaseIV [7]. Вместе с этим конвертором разработан и компилятор языка dBaseIV.

Помимо большого количества различных СУБД, рынок программного обеспечения ПЭВМ предлагает множество сопутствующих продуктов - генераторов отчетов, средств построения экранной формы ввода/вывода и меню, генераторов приложений, совместимых с СУБД, графических пакетов, обучающих программ и даже средств автоматизации проектирования баз данных. В этом изобилии трудно зачастую ориентироваться даже искушенному специалисту.

В отечественных и зарубежных изданиях опубликовано значительное количество разнообразных обзоров (см., например, [8-11]), всевозможных рекламных материалов, характеризующих функциональные возможности тех или иных СУБД для ПЭВМ, сравнительные оценки их характеристик, результатов измерений параметров производительности, сведений о проведенных рейтингах.

Эти материалы в большинстве своем отражают позицию пользователей, а не специалистов в области технологии баз данных, часто служат коммерческим интересам поставщиков конкретных программных продуктов. Сопоставление производительности систем проводится на наборах тестов, зачастую подобранных без должного научного обоснования, вследствие чего информация, представленная в таких публикациях, не всегда выглядит убедительно.

Для полноценного использования возможностей "персональных" баз данных необходим профессиональный взгляд на состояние, особенности и тенденции развития такого инструментария с позиций традиционной технологии баз данных, имеющей уже четвертьвековую историю. В результатах такого анализа нуждается и большая группа специалистов, владеющих технологией баз данных на больших вычислительных системах и мини-ЭВМ и приступающих к использованию ПЭВМ.

В предлагаемой работе предпринимается попытка такого анализа. При этом не преследуется цель провести полную инвентаризацию и расклассифицировать все то большое количество программных продуктов, из которых вынужден делать свой выбор пользователь. Мы пытаемся выявить общие характерные особенности наиболее популярных на международном рынке программных продуктов рассматриваемого класса, работающих в среде операционной системы MS-DOS, оценить тенденции развития рассматриваемого класса инструментальных средств.

В данной работе мы будем опираться в основном на характеристики таких широко распространенных систем, как

dBaseII PLUS и dBaseIV (ASHTON-TATE Corp.)  
R:base for DOS (MICRORIM Inc.)  
Oracle (ORACLE Corp.)  
Paradox 2.0 (ANSA SOFTWARE)  
db\_Vista (RAIMA Inc.)  
DataEase (DataEase INTERNATIONAL Inc.)

и многочисленных сопутствующих им программных продуктов.

## Многообразие средств управления данными

Персональные ЭВМ располагают чрезвычайно широким спектром средств управления данными - от программных продуктов, реализующих как отдельную функцию методы доступа к хранимым во внешней памяти данным, до развитых СУБД.

В виде самостоятельных продуктов реализованы методы доступа к данным, основанные на В-деревьях, и индексно-последовательный метод доступа.

Средства управления данными предусмотрены в системах "электронных таблиц" (Lotus 1-2-3, SuperCalc, Quattro), в интегрированных пакетах и средах (Framework, Symphony), в методоориентированных пакетах, таких как пакеты математической статистики и линейного программирования.

Поставляются также и СУБД самого различного характера - простейшие однофайловые системы с плоским файлом - PFS:Professional File (SOFTWARE PUBLISHING Corp.), Reflex (BORLAND INTERNATIONAL Inc.), сетевые - db\_Vista и реляционные СУБД - dBaseII PLUS, R:base for DOS, Paradox и другие. Многие из коммерческих СУБД располагают мультипользовательскими версиями, предоставляющими доступ пользователям сетей ПЭВМ к централизованно поддерживаемой базе данных.

Такое многообразие инструментальных средств позволяет выбирать средства, адекватные потребностям конкретной разработки, и тем самым обеспечивает рациональное использование вычислительных ресурсов. Эту задачу не всегда удавалось решить на больших системах, где в силу ограниченности выбора и значительно меньшей дифференциации возможностей инструментальных средств управления данными разработчик вынужден был часто либо вообще отказаться от использования типовых коммерческих средств, либо сознательно должен был идти на ущемление своих потребностей.

Для эффективного выбора подходящего инструмента в условиях такого изобилия необходима и более высокая профессиональная подготовка специалиста, принимающего решение относительно такого выбора.

## Дружественная пользовательская среда

Свойством, органически присущим инструментальным средствам "персональных" баз данных, является в высшей степени дружественный характер их пользовательских интерфейсов.

В подавляющем большинстве СУБД для ПЭВМ предусматривается интерактивный режим работы пользователей. При этом широко используются интерфейсы в стиле меню с указанием альтернативных вариантов выбора возможных действий для пользователя и способов их инициирования, с отображением текущего состояния системы базы данных и диагностикой ошибок. Развитые СУБД предусматривают при этом глобальную и контекстно-зависимую помощь пользователю в оперативном режиме. В значительной мере минимизируются действия пользователя, необходимые для инициализации часто активизируемых функций, благодаря применению функциональных клавиш клавиатуры ПЭВМ.

В некоторых СУБД предусматривается несколько уровней интерфейсов, предъявляющих различные требования к квалификации пользователей. Так, в системе dBaseII PLUS для пользователя-программиста предусмотрен интерфейс языка программирования системы. Система приглашает пользователя

ввести очередную команду, исполняет ее, запрашивает следующую команду и т.д. Кроме того, предусмотрена режим ASSIST, в котором пользователь общается с системой с помощью выбора альтернатив в последовательно разворачивающемся разветвленном иерархическом меню и спецификации различных параметров по запросу системы.

Механизмы такого интерфейса формируют в процессе работы на основе выборов пользователя соответствующие команды языка программирования системы и обращаются к системе для их исполнения и, таким образом, избавляют пользователя от необходимости знания языка программирования системы. Аналогичные функции исполняют компоненты системы Paradox - Paradox Personal Programmer, формирующий на основе выборов пользователя по предоставленному ему меню тексты программ на языке программирования системы - PAL (Paradox Programming Language), или EXPRESS - модули системы R:base.

Дружественный характер СУБД для ПЭВМ хорошо гармонирует с главной направленностью технологии персональных ЭВМ на обеспечение комфортных условий работы пользователей. Благодаря этому достигается существенно более высокая производительность труда пользователя при создании и эксплуатации "персональных" баз данных, чем при выполнении подобных работ на "больших" ЭВМ со свойственными такой социальной пользовательской среде значительными накладными расходами.

Хорошо сбалансированное сочетание простоты освоения и использования "поверхностных" функций таких массовых СУБД, как dBaseII PLUS, R:base, Paradox, с активизируемыми с помощью простого интерфейса богатыми функциональными возможностями обеспечивает этим системам широкую сферу применения, легкое и быстрое удовлетворение простых информационных потребностей. Первые благоприятные впечатления от dBaseII PLUS, например, оказываются настолько сильными, что порождают у непрофессионального пользователя иллюзии о том, что любую информационную систему можно реализовать методом самообслуживания в кратчайшие сроки и без заметных трудозатрат, что в этой области вообще не существует каких-либо проблем и какой-либо профессиональной специфики, требующей специальных знаний.

Однако по мере усложнения информационных потребностей, когда уже не удается обойтись средствами простейшего пользовательского интерфейса с системой и приходится прибегнуть к ее языку программирования, иллюзии начинают рассеиваться.

Если объемы данных в базе данных достигают значительных размеров, то при выполнении операций, требующих прямого доступа, пользователя начинает беспокоить недостаточно высокая реактивность системы. Для решения проблемы повышения производительности системы базы данных уже необходимы не только хорошее знание языка программирования системы, но также профессиональные знания в области технологии баз данных и понимание механизмов доступа к базе данных в рассматриваемой СУБД.

## Модели данных

Среди СУБД общего назначения для персональных ЭВМ имеется ряд систем, поддерживающих модель данных плоских файлов [9] (уже упоминавшиеся PFS:Professional File и Reflex, Rapid File и др.).

Однако пользователь обладает довольно ограниченным выбором среди систем, поддерживающих сетевую и иерархическую модели данных. Известна СУБД db\_Vista (RAIMA Corp.), работающая с сетевой моделью данных. Это весьма упрощенная сеть КОДАСИЛ, в которой полностью исключена автоматика в навигации по структуре базы данных и в распространении



изменений. Заметим в качестве курьеза, что в системной документации не удалось найти какое-либо упоминание о КОДАСИЛ. В обзоре [11] упоминаются также как сетевые СУБД системы Q-Pro 4 и Zim. В том же обзоре имеются краткие сведения о СУБД иерархического типа - PC/Focus, Team-Up и Data Edge. Помимо большинства СУБД для персональных ЭВМ составляют системы реляционного типа или системы, квалифицируемые разработчиками как реляционные. Здесь уместно вспомнить Э. Кодда, предложившего считать реляционной такую СУБД, которая не только поддерживает реляционную структуру данных, но и реализует, по крайней мере, операции соединения, проекции и селекции отношений [12]. Этот вопрос обсуждается также в работе К. Дейта [2].

Преобладание реляционных систем на рынке программного обеспечения может вызвать некоторое удивление. Действительно, хорошо известны те трудности, которые возникали при попытках эффективной реализации реляционной модели данных на больших вычислительных системах, работы, направленные на создание специальных аппаратных средств (реляционные специализированные машины баз данных), позволяющих достичь приемлемой производительности реляционных СУБД. Фирма IBM затратила около десяти лет на проведение серьезных исследований и разработок, прежде чем выпустила на рынок первые коммерческие реляционные СУБД SQL/DS и DB/2 [2] для своих больших вычислительных систем. Возникает естественный вопрос, каким же образом удалось преодолеть эти трудности на персональных ЭВМ? Мы полагаем, что ответ весьма прост: никаких специальных новых шагов для преодоления таких трудностей никто и не пытался предпринимать. Наша гипотеза состоит в том, что столь массовый выбор реляционной модели данных разработчиками программных средств для ПЭВМ определялся удобством этой модели для пользователя, рекламными соображениями - о достоинствах реляционной модели уже было хорошо известно, а также сравнительной простотой ее реализации, особенно если не слишком обременять себя проблемами производительности.

На ранней стадии технологии персональных ЭВМ проблемы производительности СУБД не являлись острыми, поскольку размеры баз данных были невелики. Вполне удовлетворительным решением являлось использование механизмов индексирования для обеспечения более быстрого доступа к требуемым данным. Когда впоследствии с появлением жестких дисков большого объема в десятки и сотни мегабайт эти проблемы выдвинулись на передний план, стали приниматься меры по их решению.

Указанная структура существующего парка СУБД для ПЭВМ привела к усилению негативной тенденции, которая начала складываться в технологии баз данных на больших вычислительных системах. Вопрос об адекватности используемой в данной СУБД модели данных характеру предметной области реализуемого приложения перестал играть сколько-нибудь значительную роль при выборе подходящей системы. В условиях изобилия реляционных СУБД на ПЭВМ решение этого вопроса практически почти всегда предрешено - это реляционная система.

Нужно заметить, что имеющиеся реляционные системы заметно различаются набором тех реляционных операций, которые в них реализованы. Во многих системах не предусмотрены операции пересечения, объединения и разности отношений.

Хорошо известно, что важным инструментом обеспечения логической независимости данных в системах баз данных являются механизмы представлений (view). Вызывает некоторое удивление, что в большинстве реляционных систем и даже в таких широко

распространенных СУБД, как dBaseII PLUS и Paradox, явные механизмы представлений отсутствуют. В то же время такие средства предусмотрены в R:base for DOS, Oracle, DB2. Их использование существенно облегчает разработку приложений и позволяет более экономично использовать ресурсы памяти, конечно, за счет увеличения времени обработки запросов.

## Типы данных и ограничения целостности

Следует особо остановиться на таких аспектах моделей данных в "персональных" СУБД, как типы значений данных и ограничения целостности.

Во многих системах предусматриваются числовые значения с фиксированной и плавающей точкой, с двойной точностью, строковые значения, булевые значения, значения в национальных денежных единицах, даты и отметки времени. Предусматривается также естественная арифметика дат, значений времени и денежных величин.

Большие неудобства в использовании системы dBaseII PLUS и некоторых других СУБД в научных и инженерных приложениях доставляет отсутствие в ней возможности оперировать с числовыми величинами, представленными в форме с плавающей точкой.

Важную роль в операциях с базами данных играет концепция неопределенного значения (NULL-value). Эта концепция, к сожалению, не поддерживается в ряде коммерческих СУБД для ПЭВМ. Так, в базах данных системы dBaseII PLUS нельзя отличить значение числового поля "ноль" или строку пробелов как значение литерного поля от неопределенного значения. От этого недостатка свободны системы Paradox и R:base for DOS.

В теории систем баз данных большое внимание уделяется средствам спецификации ограничений целостности данных как составной части модели данных [13]. Ограничения целостности рассматриваются при этом как механизм моделирования семантики предметной области в базах данных.

К сожалению, возможности спецификации ограничений целостности в наиболее популярных СУБД для ПЭВМ весьма примитивны. Обычно они ассоциируются не с объектами базы данных, а с полями форм ввода/вывода. Для величин, обладающих количественными типами данных (числа, даты, денежные величины), используются, как правило, ограничения, специфицирующие возможный диапазон изменения, а для литерных строк - шаблоны представления значений. Во многих реляционных СУБД не поддерживается концепция первичного ключа отношения.

Приятным исключением является система R:base for DOS. В ней не только поддерживаются первичные ключи отношений (таблиц), но и предусмотрены средства спецификации довольно сложных ограничений целостности. Их аргументами может быть множество значений атрибутов в кортеже одного или нескольких отношений. Ограничения целостности ассоциируются в системе с атрибутами отношений.

Развитые средства спецификации ограничений целостности предусматриваются и в СУБД, в полной мере реализующих язык SQL [2,14]. Для примера можно сослаться на систему DB2 фирмы IBM или dBaseV (ASHTON-TATE Corp.). Предложение ASSERTION в языке SQL предусматривается именно для этой цели.

## Особенности архитектуры "персональных" СУБД

Оценивая архитектуру наиболее популярных СУБД для ПЭВМ с позиций известной архитектурной модели ANSI/SPARC [15], можно констатировать, что мы имеем здесь дело с упрощенными в значительной мере

случаями. Упрощения касаются прежде всего внутреннего уровня архитектуры и механизмов отображения данных "концептуальный уровень - внутренний уровень".

Определяя структуру концептуальной базы данных в СУБД для ПЭВМ, проектировщик тем самым однозначно устанавливает физическую структуру базы данных. СУБД рассматриваемого класса не предусматривают возможностей непосредственного управления физической структурой базы данных. Здесь используются, таким образом, механизмы межуровневого отображения данных по умолчанию [16,17].

В отличие от больших вычислительных систем, механизмы распределения памяти операционных систем ПЭВМ не предусматривают резервирования пространства для файлов во внешней памяти, и оно распределяется динамически. В силу этого управление пространством памяти в среде хранения со стороны проектировщика системы баз данных в большинстве СУБД для ПЭВМ сводится только к указанию имен файлов, в которых будут храниться данные, и дисководов, на которых они размещаются. В реляционных СУБД чаще всего при этом отдельный файл служит для хранения одного отношения, и его имя совпадает с именем этого отношения. Каждый кортеж хранится в соответствующей записи этого файла. Такой подход использован, например, в системе *dBaseIII PLUS*.

Функцию выбора места размещения кортежей отношений в пространстве памяти среди хранения здесь по существу выполняет сам пользователь. Для динамического управления пространством памяти СУБД использует функции операционной системы, а возвращение освободившегося пространства для повторного использования осуществляется только по специальной команде пользователя.

Однако имеются и более сложные случаи. Так, в системе *R:base for DOS* в одном файле операционной системы хранятся все пользовательские данные - все отношения базы данных. При этом управление как пространством памяти, так и размещением данных является функцией СУБД.

В качестве путей доступа в *dBaseIII PLUS* и *dBaseIV*, *R:base for DOS* и *Paradox* могут использоваться: прямой доступ по "ключу базы данных" (по относительному номеру записи) кортежа в файле среди хранения, доступ с помощью индекса при заданном значении ключа индексирования кортежей, а также доступ по значениям атрибутов кортежей или по какому-либо зависящему от них предикату более сложного вида. В этом последнем случае доступ осуществляется путем последовательного перебора кортежей до тех пор, пока не удастся найти первый из них, удовлетворяющий критерию поиска.

Отметим, что в системах *dBaseIII PLUS* и *dBaseIV* система не берет на себя определение возможности и целесообразности использования механизмов индексирования для повышения эффективности доступа к данным в базе данных. Эту задачу должен решать пользователь и при необходимости явно указывать системе, что доступ должен быть осуществлен с помощью механизмов индексирования.

Возможность аппелировать в пользовательских запросах к базе данных - по существу, в функциях внешнего архитектурного уровня - к категориям среды хранения (индексы и ключи базы данных) не соответствует представлениям ANSI/SPARC о распределении функций между архитектурными уровнями СУБД и нарушает тем самым "чистоту" архитектуры рассматриваемых систем.

Если говорить о внешнем уровне архитектуры, то система *db\_VistaIII* (RAMA Corp.) [18] демонстрирует случай мультимодельного подхода, хотя и значительно упрощенный. Здесь пользователь может применить

помимо сетевой модели данных еще и реляционную, в терминах которой могут специфицироваться поисковые запросы, касающиеся части информации, содержащейся в базе данных. Указанная возможность реализуется модулем *db\_Query*. Однако в данном случае реляционный интерфейс оставляет недоступной информацию, представляемую типами и экземплярами наборов.

Основным инструментом конструирования объектов данных внешнего уровня в реляционных СУБД для ПЭВМ являются естественные для систем этого типа механизмы представлений. Чаще всего это представления, определенные над одной таблицей. В системах *DB2* фирмы IBM, *R:base for DOS* и других можно однако построить многотабличные представления.

### Пользовательские интерфейсы

Еще на ранней стадии формирования технологии баз данных была принята дилемма пользовательских интерфейсов СУБД по характеру их языковых средств - интерфейсы включающего и интерфейсы автономного (самостоятельного) языков [19]. Созданные в этот период (середина 60-х - начало 70-х годов) системы обычно обладали каким-либо одним из таких интерфейсов. В более поздних СУБД предусматривались, как правило, интерфейсы обоих типов.

Среди СУБД для персональных ЭВМ, пользуясь терминологией обзора [11], можно выделить программируемые и непрограммируемые системы. Системы последнего класса вообще не обладают каким-либо языковым интерфейсом для пользователя. Взаимодействие пользователя с системой осуществляется с помощью интерфейсов, построенных в стиле меню.

Подобные интерфейсы используются и во многих программируемых системах. Однако в этих случаях система управляется в конечном счете тем или иным языком программирования.

В преобладающем большинстве таких систем интерфейсы языка программирования построены по принципу интерфейса автономного языка. Таков *Paradox* с его языком *PAL* [20], система *dBaseIII PLUS* с ее *dBase*-языком [21,22] и *dBaseIV* с языками *dBase* и *SQL* [23].

Однако существуют и примеры другого рода, когда система обладает языковыми интерфейсами обоих видов. Так, *R:base for DOS* располагает как автономным языком программирования, так и модулем *PI* (Program Interface), обеспечивающим доступ к базам данных системы из программ, написанных на языках Microsoft Fortran, Microsoft Pascal, Microsoft C. Пакет *R:bridge* оснащен такими же возможностями аналогичные системы программирования фирмы *BORLAND INTERNATIONAL Inc.* - *Turbo Fortran*, *Turbo Pascal*, *Turbo C*.

Система *db\_Vista* имеет интерфейс включающего языка для различных компиляторов языка Си. Вместе с тем, модуль *db\_Query* обеспечивает SQL-интерфейс автономного языка для пользователя.

Фирма *LATTICE Inc.* разработала пакет *dbCII Library* [29] - библиотеку функций на языке Си, позволяющую реализовать интерфейс включающего языка Си для операций с базами данных системы *dBaseIII PLUS*. Аналогичные функции для языка Си выполняет пакет *Code Base* [24]. Фирма *AJS PUBLISHING Inc.* создала подобное средство *dbLIB* для системы программирования *Quick BASIC* [25].

Автоматные языки программирования СУБД часто не являются таковыми в полном смысле слова. Так, из программ, написанных на языках систем *R:base*, *dBaseIII PLUS* и *dBaseIV*, можно вызывать загрузочные модули, заблаговременно созданные на любом языке программирования. Обмен данными между вызывающей и вызываемой программами может при этом осуществляться через параметры командной строки,



инициирующей вызываемую программу, и ASCII-файлы операционной системы, в которые эти СУБД легко экспортируют данные из базы данных и из которых они легко их импортируют.

Нужно заметить, что все большую популярность среди разработчиков СУБД на ПЭВМ приобретает язык SQL. Он реализован в системах Oracle, dBaseIV, db\_Vista (модуль db\_Query), DB2, XQL (фирма SOFT CRAFT), dQuery (QUADBANE SYSTEMS, Inc), Informix (INFORMIX SOFTWARE Ltd.) и ряде других. Его подмножество входит в язык системы R:base for DOS. А в системе Ingres - одной из первых коммерческих реляционных СУБД, реализованной на ЭВМ PDP-11 в Калифорнийском университете (США) и перенесенной теперь на персональные ЭВМ, совместимые с IBM PC, язык SQL реализован наряду с не менее развитым "родным" языком Quel этой системы.

### Формы вывода данных и графические возможности

Для представления данных пользователю в системах "персональных" баз данных применяются прежде всего традиционные табличные формы, порождаемые генераторами отчетов. Некоторые генераторы отчетов, например в системе Paradox, позволяют выводить отчеты в свободном формате. Наконец, весьма часто используются экранные формы ввода/вывода данных, построенные с помощью экранных редакторов, входящих в состав каждой развитой СУБД. Такие редакторы имеются в dBaseIII PLUS, R:base for DOS, в системах Paradox и DB2, в системе Oracle. Для системы dBaseIII PLUS имеются другие редакторы экранных форм, как FlashCode (SOFTWARE BOTTLING Comp.), QuickCode (FOX & GELLER), которые, помимо функций редактора, еще генерируют некоторый стандартизованный исходный код на языке dBase, включающий набор процедур для ввода, редактирования и поиска данных в базе данных. Пользовательский интерфейс при этом базируется на построенной экранной форме.

В научных, инженерных и управленческих приложениях широко используются графические возможности представления данных, извлекаемых из базы данных. Для персональных ЭВМ разработаны разнообразные графические средства. Имеются многочисленные графические пакеты, а также инструментарий, встроенный в системы программирования. Богатыми возможностями обладает, например, система программирования Turbo Pascal (BORLAND INTERNATIONAL Inc.).

Языки программирования систем dBaseIII PLUS, DataEase (DataEase INTERNATIONAL Inc.) или R:base for DOS позволяют вызывать заранее скомпилированные и отредактированные модули на любом традиционном языке программирования. Поэтому приложения, реализованные их средствами, могут обратиться к написанной на языке Си, Турбо-Паскаль или каком-либо ином языке программе, формирующей графическое представление выходных данных. А эти данные они могут получить от вызывающей программы в ASCII-файле, в который СУБД их легко экспортирует.

Однако более интересны программные продукты, непосредственно работающие с данными, хранимыми в базе данных. Такими возможностями обладают пакеты dGraph for dBase (FOX & GELLER) для dBaseIII PLUS, DB Graphics (MICRORIM Inc.) для различных версий системы R:base - R:base4000, R:base 5000, R:base System V, а также для dBaseIII PLUS [26], пакет dGE (BITS PER SECOND Ltd.) - для dBaseIII PLUS и dBaseIV, Clipper, Foxbase+, QuickSilver и dBFast [24]. Собственным, совместимым с ним пакетом деловой графики DataEase GrafTalk, обладает система DataEase. Деловая графика предусмотрена также в системе Paradox 3.0 [27].



1. Oracle. Database Administrator's Guide. Oracle Corp. 1984.
2. Дейт К. Руководство по реляционной СУБД DB2. - М.: Финансы и статистика, 1988.
3. Byte, September 1989.
4. Using dBaseIII PLUS. Ashton-Tate Corp., 1986.
5. PC World. April 1988, p.255.
6. PC World. February 1989, p.84.
7. Learning dBaseIV. Ashton-Tate Corp., 1988.
8. Сенин Г.В. Системы манипулирования данными на персональных ЭВМ//Прикладная информатика. - М.: Финансы и статистика, 1988. - Вып.14 - С.91-118.
9. PC Magazine, April 12, 1988.
10. PC Magazine, April 26, 1988.
11. PC Magazine, May 17, 1988.
12. Codd E.F. Relational Database: A Practical Foundation for Productivity. Commun. of ACM, 25, no.2, February 1982.
13. Цикритзис Д., Лоховски Ф. Модели данных. - М.: Финансы и статистика, 1985.
14. Chamberlin D.D. et al.: SEQUEL-2: A unified approach to data definition, manipulation, and control. - IBM J. Res. & Dev., 20: 6, 1976, p.560-575.
15. ANSI/X3/SPARC Study Group on data base management systems: Interim report. FDT, 7:2, ACM, New York, 1975.
16. Когаловский М.Р. Архитектура механизмов отображения данных в многоуровневых СУБД//Техника реализации многоуровневых систем управления базами данных. - М.: ЦЭМИ АН СССР, 1982. - С.3-19.
17. Когаловский М.Р. Механизмы отображения данных по умолчанию в многоуровневой СУБД//Математические и реализационные проблемы СУБД. - М.: ЦЭМИ АН СССР, 1981. - С.87-95.
18. db\_Vista User's Guide. Raima Corp., 1986.
19. Информационные системы общего назначения/Пер. с англ.; Под ред. Е.Л. Ющенко. - М.: Статистика, 1975.
20. The PAL User's GUIDE. The Paradox Application Language. Release 2.0. Ansa Software, 1987.
21. Programming with dBaseIII PLUS. Ashton-Tate Corp., 1986.
22. Liskin M. dBaseIII PLUS. The Pocket Reference. McGraw-Hill, 1988.
23. dBaseIV Language Reference. Ashton-Tate Corp., 1988.
24. PC World. January 1989, p.91.
25. PC Magazine. October 11, 1988, p.386.
26. DB Graphics. Version 1.0. September 1987. Microrim Inc., 1987.
27. PC World, February 1989, p.72.
28. dBFast For dBaseIII PLUS. Reference Guide. dBFast Inc., 1988.
29. Lattice dBCII Library for MS-DOS. Lattice Inc., 1986.
30. Верещанский Д.Г. Практическое программирование на dBase. - М.: Финансы и статистика, 1989.
31. Крамм Р. Системы управления базами данных dBaseII и dBaseIII для персональных компьютеров. - М.: Финансы и статистика, 1988.

В мире "Компьютера"

© Ежи Оркишевский

# Новинки

Силиконовая долина (Silicon Valley) - это условное название южной части Бэй-Эриа, расположенной недалеко от Сан-Франциско. Название охватывает целый ряд населенных пунктов, которые объединяются в гигантский городской организм, прорезаемый автострадами, проспектами и улицами.

мк

А вдоль, среди кедров, кактусов и пальм разбросаны миллионы одноэтажных, а в лучшем случае трехэтажных зданий. Почему так низко? Потому что это сейсмическая зона, о чём госпожа Природа напомнила 17 октября 1989 г. Однако и люди, и постройки были к этому подготовлены. Последний толчок силой 7,1 баллов практически не прервал работу Силиконовой долины.

О том, что это сердце американской электронной промышленности, напоминают не только таблицы с названиями компьютерных гигантов, но и названия улиц. Известная фирма Seagate расположилась на улице... Диск-драйв (это можно переводить как "дисковод" и "дисковый проезд"), неподалеку от перекрёстка с Семикондактор-драйв ("полупроводниковый проезд"). Даже гигантские рекламные щиты, стоящие вдоль автострад, рекламируют наравне с "Кока-Колой" и автомобилями "Ниссан" новый микропроцессор Intel 80386SX и новую версию Norton Utilities.

На первый взгляд может показаться, что жизнь в Силиконовой долине течет спокойно и даже шикарно. Однако это только кажущееся спокойствие: в действительности здесь не прекращается борьба за господство на рынке. И лучше всего это иллюстрируют рекламы, которые либо публикуются в многочисленных профессиональных журналах, либо попадают в виде листовок в домашние почтовые ящики - во все подряд, без разбора.

Например, фирма INTEL яростно сражается за успех микропроцессора 80386SX, упрощенной версии процессора класса 386. Но

Из долинки





ый микропроцессор снабжен 16-разрядной шиной данных и рекламируется как 286, способный работать под контролем программного обеспечения для 32-разрядных процессоров (систем XENIX 386 или PC-MOS 386). Фирмы, выпускающие быстродействующие процессоры 286, немедленно снизили цены на свои изделия и пытаются обогнать INTEL. Например, фирма ADVANCED MICRO DEVICES доказывает, что их 286 (16 МГц) гораздо быстрее и в два раза дешевле!

Кроме того, у фирмы INTEL есть и другие трудности, связанные с незамеченной вовремя ошибкой в ее новейшем продукте - микропроцессоре 80486. Как на зло, повторилась история 80386: снова в АЛУ прокралиась ошибка, снова первые компьютеры с "неправильным" процессором были проданы, снова (!) ошибку обнаружили конструкторы из фирмы COMPAQ. Представители INTEL стараются преуменьшить значение ошибки, однако известно, что поставки микропроцессоров приостановлены по крайней мере на два месяца. А объяснения представителя фирмы, что ошибка настолько незначительна, что потребует изменений лишь в полутора десятках из 1,2 млн. транзисторов, никого не убедили.

Большие неприятности из-за этой ошибки были у фирмы IBM, которая успела выпустить на рынок компьютер с таким микропроцессором - Power Platform 486. Представитель IBM заявил, что со временем концерн заменит новыми все микропроцессоры, а до этого момента посоветовал пользователям "применять программы, которые не требуют каких-либо вычислений". Этот "добрый" совет вызвал лавину иронических комментариев со стороны конкурентов.

Потери, вызванные неудачей с микропроцессором 80486, INTEL намеревается восполнить в новой для себя области программного обеспечения. Генеральный директор фирмы по вопросам систем Димитри Ротоу (Dimitri Rotow) заявил, что INTEL собирается разработать сокращенную версию UNIX для всех компьютеров с микропроцессорами этой фирмы. Система, не имеющая еще названия, будет продаваться по цене MS-DOS. В настоящее время, - сказал Ротоу, - 300 тыс. компьютеров с нашими микропроцессорами работают с UNIX, а 27 млн. - с MS-DOS. Мы постараемся изменить эти пропорции". На вопрос, будет ли INTEL расширять производство программного обеспечения, он ответил, что новая система - лишь часть работы, связанный с внедрением микропроцессора 80486, и добавил: "Мы не намерены конкурировать с фирмами MICROSOFT или BORLAND, но 80486 стоил нам свыше миллиарда долларов".

Гигантам вообще живется тяжело. Попытка фирмы HEWLETT-PACKARD ввести новый стандартный язык, предназначенный для управления измерительными системами, закончилась провалом. Уже в самом названии языка было слово "стандарт" (TMSL - Test and Measurement Standard Language). Но конкурирующие фирмы, в том числе TEKTRONIX и KEITHLEY, объединили силы и вводят теперь свой "стандарт". MICROSOFT тоже "не спит". Билл Гейтс сказал, что начиная с 1990 г. будет производить только программы для графической среды Windows.

IBM готовит для семейства PS дисководы оптических дисков "только для чтения". Они будут снабжены драйвером, совместимым с сетью Token Ring. Их предполагаемая цена не будет превышать 1000 дол. Для кого же они предназначены? По-видимому, прежде всего для школ, так как на оптических дисках CD-ROM записан уже почти весь материал, предусмотренный стандартной программой обучения.

А что из новинок можно будет в ближайшее время увидеть у нас, на востоке Европы? Несомненно, системы, основанные на микропроцессоре 80486. Среди

них лучшие рецензии получает Compaq Systempro. Эта полная новаторских решений машина, программно-совместимая с XT/AT/386 и использующая "обычные" расширяющие карты, работает на 2 центральных процессорах. Это могут быть либо два микропроцессора типа 386, либо два типа 486, либо 386 и 486. Один из них находится на базовой плате, а второй - на 32-разрядной карте, подсоединенной к соответствующему разъему. В корпусе типа tower можно разместить до 8 жестких дисков общей емкостью 1,68 Гбайта. ОЗУ может насчитывать от 4 до 256 Мбайт. Испытания показали, что два процессора типа 486 работают с мощностью порядка 40 млн. команд/с. При использовании этого ПК как файл-сервера, к которому "подключено" 60 активных пользователей, оказалось, что он работает в 6 раз быстрее, чем мини-ЭВМ VAX 6310, и (в зависимости от режима работы) в 6-38 раз быстрее, чем мини-ЭВМ IBM AS400. Если кого-либо интересует эта машина, то сообщаю, что в самой дешевой конфигурации с жестким диском 240 Мбайт ее можно купить за 15999 дол., а в полной версии с жестким диском 420 Мбайт - за 25499 дол. Дополнительно можно применить интеллектуальный контроллер жесткого диска с микропроцессором Motorola 68000 и ОЗУ 4 Мбайта. Это снижает среднее эффективное время доступа к диску до (внимание!) 0,6 миллисекунды. Но это для тех, кто очень торопится. Торговые фирмы не перегонки выдумывают, чем привлечь новых клиентов. Например, в конце прошлого года президент фирмы VPC (Very Important Person Computer) Тони Коул предложил своим клиентам несколько компьютеров, в которых было все "самое-самое". А дополнительно вся электроника была упакована в... позолоченные корпуса! Цена такой машины составляла 50000 дол., в том числе золота было на 10000 дол. Говорят, что за ними стояла очередь...

Но пусть это никому не портит самочувствия. Средний американец работает на коши IBM, изготовленной одной из фирм на Тайване или в Сингапуре. То есть все нормально!

Перевод Анджея Поплавского



# Компьютеризуемся

## ELBkey – РЕШЕНИЕ ПРОБЛЕМ

© Андрей Сучков

Несмотря на героические усилия, прилагаемые отдельными гражданами, кооперативами и некоторыми другими организациями по части всеобщей компьютеризации, рабочие места, оборудованные персональными компьютерами индивидуального пользования, по-видимому, еще на долгие годы останутся лишь мечтой для большинства наших соотечественников. Осознание персонального компьютера как инструмента, совершенно необходимого в повседневной работе, в условиях дефицита самих персональных компьютеров порождает парадоксальную ситуацию – персональный компьютер перестает быть персональным в изначальном смысле этого слова. Рядовой пользователь, получивший доступ к ПК, вынужден "делить" его с коллегами, сблюшая некоторую очередь в работе. Типичная проблема, с которой он при этом рано или поздно сталкивается, – защита своих файлов, хранящихся на жестком диске.

Прочитав описание операционной системы и с огорчением убедившись, что разработчики MS-DOS, по-видимому, никак не рассчитывая на подобный "многопользовательский" режим работы, не предусмотрели встроенных средств защиты, вы начинаете искать другие возможности. Первое, что, естественно, приходит в голову начинающему, – это хранить файлы своих дан-

ных и программ на дискетах, переписывая их на жесткий диск только на время сеанса работы. Узнав, как легко с помощью специальных (но отнюдь не труднодоступных) средств могут быть восстановлены, а значит, и прочитаны удаленные с винчестера файлы и испытав горечь поражения, вы переходите к исполь-



зованию более сильных средств – программ-архиваторов или программ, специально предназначенные для ограничения доступа к файлам пользователя. Программы-архиваторы, такие, как PK35A35 (PKWARE Inc.), PAK (NOGATE CONSULTING) или входящая в состав популярного пакета PC SHELL (CENTRAL POINT SOFTWARE) программа перекодировки PCSECURE – не действитель обес печивают парольное кодирование и надежную защиту ваших файлов, но... лишь до тех пор, пока эти файлы не понадобились вам для работы, так как декодированные перед началом работы программы и данные записываются на жесткий диск в своем первозданном виде. PCSECURE при этом умудряется "раскрыть" не только затребованные вами файлы, но и все остальные, связанные с этим паролем и находящиеся в том же подкаталоге. Еще один серьезный недостаток подобных программ, про-

являющийся особенно явно при их систематическом применении – невозможность использования средств защиты непосредственно во время сеанса работы с другими прикладными программными системами. Пользователь, же лающий, например, редактировать текстовый файл, предварительно обработанный программой PCSECURE или архиватором, должен прекратить работу с текстовым процессором, выйти в операционную систему, выполнить действия, необходимые для декодирования файлов, снова вернуться в редактор и только после этого приступить непосредственно к работе. К старым проблемам добавляются новые...

Каким же средством можно воспользоваться для защиты своих программ и данных? В начале этого года научно-производственное объединение ЭЛВИ (г. Москва) объявило о выпуске своего нового программного продукта – программного средства ELBkey, предназначенного для обеспечения санкционированного доступа к программам и данным на персональном компьютере. ELBkey обеспечивает парольное кодирование любых файлов пользователя на гибких дисках и винчестере. Дальнейший доступ к закодированным файлам возможен лишь при наличии специальной резидентной программы – ключа, работающей в соответствии с паролем, который сообщается пользователем при ее запуске. Ключ "прозрачен" для большинства программных средств, работающих под управлением операционной системы MS-DOS версии 3.x и

выше, что позволяет выполнять декодирование защищенных файлов одновременно с их загрузкой в оперативную память. На дисках файлы хранятся в защищенном виде. Программное средство ELBkey не только позволяет надежно защитить программы и данные от несанкционированного доступа, но и



поддерживает много-пользовательский режим работы. Число пользователей определяется вариантом поставки и может быть изменено, а кодирование файлов уникально для каждого из пользователей во всей совокупности поставок средства. Вместе с тем ELBkey не препятствует работе пользователей, не включенных в систему санкционирования доступа, с их программами и данными. Работающий с ELBkey чувствует себя комфортно и не нуждается в специальной подготовке – разработчиками предусмотрено меню, позволяющее легко перемещаться по дискам и каталогам, осуществлять селективный выбор файлов, получать справочную информацию. Все, что требуется для эксплуатации программного средства, – это персональный компьютер минимальной конфигурации. Нет сомнения, что эта разработка НПО ЭЛВИ заинтересует всех пользователей, нуждающихся в защите своих программ и данных.

По всем вопросам, связанным с получением дополнительной информации и приобретением ELBkey, следует обращаться в редакцию журнала "Компьютер".





## ЗАЩИТА ФАЙЛОВ

© Сергей Груздев

На рынке программного обеспечения появилась программа File\_PROTECTION – принципиально новое средство профилактики и борьбы с компьютерными вирусами.

Задача системы – известить вас о "заболевании" запущенной программы, восстановить "зараженную" программу независимо от типа вируса, позаботиться о сохранении ваших авторских прав, предотвратив незаконное копирование и тиражирование ваших программ, обеспечить заданное вами количество копий (инсталляций) программы при поставке ее заказчику. File\_PROTECTION избавит вас от создания рекламной версии системы, обеспечив свободное распространение вашей программы на дискетах, одновременно блокируя ее работу на винчестере.

Сфера применения – защита от нелегального копирования и от заражения вирусами для любых исполняемых файлов (в форматах .EXE и .COM) в среде MS-DOS (PC-DOS) версии 3.00 и старше.

В системе File\_PROTECTION разработан и реализован принципиально новый метод проведения профилактики и борьбы с компьютерными вирусами. Суть метода – в предварительной "вакцинации" программы, т.е. искусственном "заражении" программы разработанной нами вакциной (по сути своей являющейся вирусом). При запуске программы управление получает искусственно дописанный в нее модуль, который проверяет, не заражена ли ваша программа. В отличии от всех антивирусных программ, настроенных только на определенные типы вирусов, для нашей вакцины тип и размер вируса не играют какой-либо существенной роли

– ей важен лишь факт заражения (естественно при условии, что не испорчен код самой программы, хотя тогда нужно говорить уже не о заражении, а об "убийстве" программы этим вирусом). Восстановление программы в этом случае не представляется возмож-



ным, как, впрочем, и для остальных антивирусных средств.

После тестирования программы на наличие в ней вируса производится ряд проверок, определяющих, является ли данная копия программы легальной. Установив, что данная копия программы является незаконной, модуль прекращает работу, так и не передав управления основной (защищенной) части программы. Кроме того, предоставляется ряд дополнительных возможностей, связанных с ограничением тиражирования и количества запусков защищенного программного продукта, что позволит сохранить ваши авторские права.

Прямых аналогов предлагаемого метода и его реализации нет. Имеет смысл говорить лишь об отдельно взятых функционально сходных программах или системах, т.е. об антивирусных пакетах и программных средствах защиты от копирования.

Что касается антивирусных программ, то в ходу их довольно большое количество, часто повторяющихся и дублирующих друг друга или являющихся их комбинацией. Все они, как правило, сканируют диск в поисках зараженных файлов по тем или иным признакам или вхождениям серии ключевых кодов в тело программы. На

это тратится довольно много времени (на комплексную проверку винчестера различными антивирусными программами до десятков минут). Терпения на такую проверку перед каждым сеансом работы хватает, к сожалению, далеко не у всех. И большинство пользователей, уже будучи не раз наказанными, продолжают следовать принципу: "Пока гром не грянет – мужик не перекрестится".

Эффективность же нашей вакцины – в диагностике на ранней стадии "заболевания", когда вирус еще находится в стадии "инкубационного периода" и не успел ничего натворить (разве что только размножиться). Вот тут-то и начинается настоящее поле деятельности для имеющихся у вас антивирусных пакетов! Но не исключено, что они ничего не найдут – у вас новый тип вируса! Клинический случай... Единственное средство спасти ценные программы – восстановить их с помощью File\_PROTECTION!

Система защиты файлов File\_PROTECTION является собственностью Начально-технического Центра "Бумеранг" и поставляется всем заинтересованным организациям, фирмам, кооперативам и частным лицам с ограниченным правом тиражирования, а также без права тиражирования.

По вопросам, связанным с получением дополнительной информации и приобретением системы File\_PROTECTION, следует обращаться в редакцию журнала "Компьютер".

## Компьютерная секция и Компьютерное общество

В июне 1989 г. прошла учредительная конференция, а в ноябре этого же года состоялась регистрация новой общественной организации – Союза ученых СССР. Союз ученых СССР является добровольным независимым творческим союзом, деятельность которого в по-

следнее время стала активно разворачиваться. В рамках Союза ученых образована Компьютерная секция. Предполагается, что в ее работе будут принимать участие:

\* специалисты, применяющие научные методы для создания компьютерных систем и их компонентов;

\* ученые, активно использующие компьютерные системы в своей профессиональной деятельности;

\* исследователи, занимающиеся социальными, психологическими, экономическими, юридическими и другими аспектами компьютеризации.

На базе секции планируется создание Всесоюзного Компьютерного общества, членами которого смогут стать вступившие в Союз ученых СССР лица указанных выше категорий. Деятельность Компьютерного общества будет включать в себя следующие основные направления:

\* издание научных и научно-практических журналов, профессиональных газет, прочая издательская деятельность;

\* организация контактов советских и иностранных специалистов;

\* проведение научных мероприятий (конференций, семинаров и т.д.);

\* подготовка законо-проектов, относящихся к сфере компьютеризации, для передачи их в высшие законодательные органы;

\* финансирование перспективных проектов;

\* разработка стандартов, соглашений, требований и т.п., связанных с построением и использованием компьютерных систем;



\* участие в комплектовании научно-технических библиотек литературы по компьютерной тематике;

\* проведение независимых экспертиз;

\* другие виды деятельности, предусмотренные Уставом Союза ученых СССР.

Всем лицам, желающим принять участие в работе секции (независимо от их членства в Союзе), а также создании и последующей работе Компьютерного общества, предлагается обращаться по адресу: 129337, Москва, а/я 31, Компьютерная секция Союза ученых СССР.



### ИРИС - инструмент для программиста

© Сергей Бебчук, Владимир Соркин

Работа с документами, как бы мы к этому ни относились, является основой повседневной деятельности служащих любого учреждения: от аппарата министерства до администрации небольшого предприятия. Можно ли использовать указанное обстоятельство при создании автоматизированных учрежденческих систем и какие требования оно предъявляет к таким системам?

Можно и необходимо - считают авторы пакета программ ИРИС (Инструмент Разработчика Информационных Систем).

Необходимо, поскольку способ взаимодействия служащего с автоматизированной системой, ориентированный на работу с документами, привычнее и, следовательно, удобнее. В идеале пользователь может рас-

сматривать информационный фонд системы как множество электронных образов бумажных документов и все операции с ними могут им мыслиться аналогично соответствующим ручным операциям. Вот модель, которая приближает пользователя к этому идеалу.

Центральным объектом в такой модели является документ. Как правило, учрежденческие документы имеют весьма сложную структуру. В них встречаются списки однотипных реквизитов, группы реквизитов, текстовая информация. Пользователь работает с внешней формой документа, которая представляет собой привычную "картиночку" - бланк, состоящий из строк определенного формата. Строки документа содержат неизменяемые поля (названия реквизитов документа, поясняющие надписи и т.п.) и слоты (графы, предназначенные для ввода информации).

Работая с документом, вы можете просматривать его и редактировать, т.е. вводить или изменять содержимое слотов. При этом можно произвольно переходить от одного слота к другому, пролистывать документ, менять порядок заполнения слотов и т.д. Иначе говоря, пользователю предлагается привычный "мягкий" режим редактирования документа, при котором порядок действий не фиксирован.

Заполненные или исправленные документы можно поместить в "картонку" - базу документов. Особенно важно при этом иметь возможность удобно и быстро находить нужные документы. Например, таким образом: вы заполняете документ-образец и получаете (почти мгновенно) множество "похожих" на образец документов. Их можно "извлечь" для просмотра и корректировки. Вам доступен также набор операций над документами: удаление, копирование или перемещение из одной базы в другую, печать и т.п. Эти опе-



рации можно выполнить сразу со всеми найденными документами.

Практически любая сложная система содержит в своей основе описанные (привычные пользователю) операции. А такие средства, как контроль значений, автоматические вычисления в одном и нескольких документах, использование подсказок и т.п. - не имеющие прямых аналогов при работе с ручной картотекой - легко описать как некоторый дополнительный сервис, предоставляемый системой за счет автоматизации.

Так, в общих чертах, должны выглядеть прикладные системы в рамках предлагаемого авторами ИРИС подхода. Создание таких систем с помощью широко распространенных СУБД реляционного типа требует достаточной изощренности: слишком существенны различия между сложной структурой учрежденческих документов и табличной формой представления данных. Дать программисту адекватный инструмент для создания таких документо-ориентированных информационных учрежденческих систем - вот цель авторов ИРИС.

Насколько удобен описанный подход могут судить пользователи уже разработанных с помощью ИРИС прикладных систем.

ПОМОЩНИК - информационно-поисковая система для работы с разнообразной деловой информацией. Она позволяетвести телефонный справочник, ежедневник и деловой блокнот, обеспечивая учет материальных ценностей, регистрацию входящих и исходящих бумаг, контроль поручений и многое другое.

ВЗЛЕТ - автоматизированная информационная система для сотрудников городских или районных органов управления, исполнков местных советов, партийных комитетов.

БИОИНФОРМ - справочная система, хранящая сведения обо всех публикациях в СССР в области биологии.

ИРИС-СКЛАД - автоматизированное рабочее место работника склада, обеспечивающее регистрацию приема и отпуска материалов.

Еще несколько десятков различных систем работают в самых разных учреждениях.

Как выглядит процесс создания прикладных систем с использованием ИРИС?

Прежде всего разработчик должен сконструировать все используемые в системе документы. Конструируя документ, разработчик "рисует" на экране внешнюю форму документа непосредственно в том виде, в котором она будет затем применяться, указывает имена и характеристики элементов, определяет их соответствие полям базы документов и может также задать для каждого элемента набор программ, обрабатывающих этот элемент.



Далее, если необходимо, разработчик может написать программы дополнительной обработки, осуществляющие специальные операции над документами и их элементами. Языком манипулирования данными ИРИС служит язык программирования Си, дополненный специализированной библиотекой функций. С точки зрения програм-



миста объекты манипулирования в ИРИС - это документы и их элементы.

Программирование прикладных систем с использованием ИРИС - это написание функций, которые осуществляют дополнительную нестандартную обработку документов, требуемую в конкретном приложении. Такие функции вызываются стандартным монитором ИРИС, который управляет универсальными процессами ввода документов, их поиска и корректировки. Наличие монитора, обладающего удобным интерфейсом для подключения функций дополнительной обработки, позволяет "локализовать" усилия разработчика прикладных систем. Разработчику не нужно программировать всю работу системы. Он должен создать только эти функции, что, безусловно, упрощает задачу.

Для создания указанных функций разработчик может воспользоваться библиотекой, которая состоит из двух частей. Первая часть содержит базовые функции ИРИС (доступ к документам и их элементам, стандартные операции над ними). Обращение к этим функциям - это обращение к самому монитору ИРИС, что позволяет сократить объем

занимаемой оперативной памяти и делает ИРИС полностью открытой системой. Вторая часть - функции, реализующие дополнительную обработку (контроль значений, автоматические вычисления значений элементов и т.д.).

Рассмотрим более подробно механизм взаимодействия монитора ИРИС с программами дополнительной обработки. Прежде всего, такие программы могут быть привязаны к документам и их элементам и загружаются в оперативную память динамически, в момент инициализации документа. Для каждого элемента можно написать набор программ, которые будут вызываться монитором при определенных событиях, связанных с данным элементом. Такими событиями являются: для документа в целом - инициализация, чтение из базы данных, запись в базу и т.д., для отдельного элемента - попадание курсора в соответствующий слот, нажатие пользователем клавиши, выход курсора из слота и т.д. При наступлении некоторого события управление передается программе, отвечающей за данное событие.

Использование языка Си дает прикладному программисту практи-

чески неограниченные возможности при создании приложений. С другой стороны, нельзя игнорировать тот факт, что многим разработчикам требуется более простой язык программирования. Возможно, со временем такой язык для ИРИС будет разработан.

Ранее созданные функции дополнительной обработки могут подключаться к элементам при конструировании документа ИРИС. Следовательно, в ряде случаев создание приложений вообще не требует программирования: класс таких приложений может расширяться путем пополнения библиотеки функций, реализующих дополнительную обработку.

Конструировать документ может (после получения некоторого опыта) и сам пользователь, особенно в тех случаях, когда структура документа полностью соответствует его внешней форме и отображению на базу документов. Таким образом, важной особенностью ИРИС является возможность (непрограммная) создания несложных приложений самими пользователями. Для программиста это дает также удобное средство взаимодействия с пользователем в ходе проекти-

рования системы, поскольку достаточно быстро могут быть согласованы формы документов.

Некоторые характеристики ИРИС помогут читателю более полно представить себе возможности пакета. ИРИС поддерживает неограниченное число файлов базы документов, в каждом файле может храниться до 2 млрд. документов, размер документа - до 64 Кбайт. Каждый документ представляет собой иерархическую структуру. В качестве его элементов могут использоваться данные всех определенных в Си типов, а также текст, дата, время и др. В каждом файле может быть создано до 255 индексируемых полей.

В заключение нам хотелось бы добавить, что первая версия ИРИС была создана еще в 1985 г. И.А. Баскиным, С.А. Вебчуком и В.М. Винц. В настоящее время работы по распространению и поддержанию текущих версий, а также по разработке новых версий ведутся в Совместном советско-франко-итальянском предприятии "Интерквадро". В момент написания данной публикации заканчивается разработка очередной версии пакета - ИРИС 2.0.

## Компьютеры и маркетинг

База данных (БД) адресов более 22 тыс. предприятий и организаций Советского Союза, имеющих службы научно-технической информации, в формате dBase для IBM PC, с возможностью сортировки по 10 типам ведомств и региональному признаку может быть использована для автоматизированной подготовки рассылки информационных материалов, эффективного маркетинга и рекламы.

Информация о возможности приобретения БД может быть представлена редакцией сборника "Компьютер" по письменному запросу организаций и предприятий.

- Мониторы...
- ...и адаптеры
- Карта телефонного модема
- Сети NOVELL
- Алиса+
- Программы-упаковщики
- Компьютерный склад информации
- Выбор принтеров
- Принтер STAR LC-15
- На книжную полку
- Пишем файл CONFIG.SYS
- Norton Commander 3.0
- Антивирусные программы



Компьютер на работе\

© Юрий Кобленц-Мишке

## Мониторы...

Видео ПЭВМ состоят из двух частей: монитора и адаптера. Заметим, что в отличие от дисплеев больших ЭВМ клавиатура ПК не относится к видео. Она подключается к компьютеру не через дисплей, а непосредственно.

Пользователь видит только монитор - похожий на телевизор прибор, который обычно стоит на системном блоке компьютера. Адаптер спрятан в корпусе машины. На экране монитора воспроизводится видеосигнал, поступающий от компьютера, видеомагнитофона, телевизионного тюнера и т.п. В самом мониторе находится только электронно-лучевая трубка и схемы развертки, реже другое устройство отображения, построенное на основе жидкокристаллической, плазменной или электролюминесцентной панели. Его

можно сравнить с колонками, которые бесполезны без приемника, проигрывателя, магнитофона.

В адаптере содержатся логические схемы, выдающие видеосигнал. Электронный луч "пробегает" экран примерно за 1/50 долю секунды, но изображение на экране компьютера меняется довольно редко и, как правило, не полностью. Поэтому видеосигнал, поступающий на монитор, должен снова и снова порождать (регенерировать) одно и то же изображение. Для его хранения в адаптере имеется видеопамять.

Каждому участку видеопамяти соответствует своя область кадра. Изображение формируется в этой памяти центральным процессором компьютера, программным путем. В некоторых адаптерах есть специальный графический сопроцессор, способный разгрузить центральный процессор от этой работы; впрочем, такие платы пока еще очень дороги, изготавливаются в малом количестве и плохо поддерживаются стандартным математическим обеспечением.

Центральный процессор имеет к видеопамяти точно такой же доступ, как и к основной памяти машины. Благодаря этому несложное изображение можно формировать на ПК очень быстро - в тысячи раз быстрее, чем на традиционной ЭВМ, соединенной с дисплеем медленным интерфейсом. Мощность машины не играет решающей роли.



Обычно ПК укомплектовывается видеоподсистемой при покупке, но заменить устаревшее устройство можно и прямо у пользователя на столе. Для этого не нужны ни высокая квалификация, ни сложные инструменты для смены адаптера, расположенного на отдельной печатной плате, достаточно обычной отвертки или ножа. Однако в последние годы адаптер встраивают в основную плату ПК. Это экономит место и удешевляет машину, но такой адаптер почти невозможно поменять на более современный.

Монитор и адаптер должны быть совместимы, но это вовсе не означает, что они обязаны жестко соответствовать друг другу. Напротив, большинство адаптеров способно работать с мониторами нескольких типов, но не всегда в оптимальном режиме.

Различные фирмы выпускают несколько сотен видов адаптеров и аналогичное число мониторов. Такое разнообразие может легко смутить покупателя, тем более советского. Невозможно, да и нет смысла перечислять все имеющиеся в продаже устройства, но мы постараемся хотя бы разобраться в терминологии и научиться различать основные классы видео.

## Основные характеристики мониторов

Прежде всего выделяются монохромные (одноцветные) и цветные мониторы. Далее мониторы отличаются разрешением. Наконец, они подразделяются на RGB и композитные, а также на аналоговые и цифровые. Особый класс образуют многочастотные мониторы ("мультисинк"). Указанные характеристики мониторов имеют принципиальное значение: они определяют совместимость с адаптером, режим его работы и возможность использования конкретных программ.

Разрешение монитора измеряется количеством строк и числом элементов изображения (сокращенно "пиксел"), а проще говоря, точек в строке. Оно обозначается формулой  $H \times V$ . Например, на мониторе с разрешением  $720 \times 348$  отображается 348 строк по 720 точек в строке. Практически все профессиональные мониторы имеют разрешение  $640 \times 200$  и более. В настоящее время чаще всего встречаются мониторы с разрешением от  $640 \times 350$  до  $720 \times 480$ ; разрешение наилучших массовых мониторов равно  $800 \times 600$  или даже  $1024 \times 768$ . Монохромные мониторы с более высоким разрешением нередко используются в настольных издательских системах. Существуют и цветные мониторы с более высоким разрешением, но они чрезвычайно дороги и применяются в основном для графических работ.

Луч монитора обычно пробегает строку за строкой, слева направо и сверху вниз (горизонтальная и вертикальная развертка), а затем возвращается к верхней строке кадра. Частота, с которой луч пробегает весь экран, называется частотой кадров, или частотой вертикального сканирования, и обычно равна 50-70 Гц. Частота, с которой выводятся строки, называется частотой строк; она примерно равна числу строк в кадре, умноженному на частоту кадров, и у подавляющего числа мониторов лежит в пределах 15-40 КГц. Наконец, частота, с которой на экран выводятся точки, т.е. с которой адаптер переключает видеосигнал, примерно равна числу точек в строке, умноженному на частоту строк, и составляет десятки мегагерц. В то время пока электронный луч возвращается к началу следующей строки (при горизонтальной развертке) и к вершине кадра (при вертикальной), на экран ничего не выводится. Поэтому частота строк и частота сигнала несколько выше, чем указанное произведение.

Изредка в мониторах применяется чересстрочная развертка, используемая в обычных телевизорах: сперва выводятся все нечетные строки кадра, а затем луч возвращается на верх экрана и высвечивает четные строки. В таком режиме примерно вдвое снижается частота строк и сигнала, но видно мигание изображения. Однако это мигание воспринимается слабее, если соседние строки кадра мало отличаются друг от друга.

Известно, что близко расположенные красную и зеленую точки одной яркости глаз воспринимает как желтую, а синюю и красную - как фиолетовую. Белый цвет представляется суммой красного, зеленого и синего цвета; при снижении их яркости он переходит сначала в серый, а затем в черный. Изменяя яркость точек, можно изменить оттенок воспринимаемого глазом цвета.

На самом деле каждый цвет можно разложить на сумму основных цветов - красного, синего и зеленого - соответствующей яркости. На этом принципе основана работа цветных мониторов (и телевизоров). Экран цветного кинескопа покрыт фосфором трех цветов. Участки каждого цвета расположены либо в виде треугольного "паркета", либо перемежающихся узких полосок; шаг покрытия составляет около  $1/3$  мм. Каждый участок возбуждается своим электронным лучом, однако все три луча движутся синхронно и всегда освещают соседние точки.

При управлении монохромным монитором видеосигнал должен нести информацию об уровне яркости каждой точки экрана, а при управлении цветным монитором - об уровнях яркости трех основных цветов, образующих пиксель.

Различия между RGB и композитными мониторами связаны с их сопряжением с адаптером. RGB-мониторы получают сигналы яркости трех основных цветов по отдельным проводам (красный, зеленый и синий по-английски red, green и blue - сокращенно RGB). Композитные мониторы получают все три сигнала по одному каналу, как в обычном телевизоре. Другими словами, сначала три сигнала объединяются в адаптере в один, а затем, уже в мониторе, вновь разделяются. Очевидно, что объединение и разделение сигналов вносит помехи, поэтому композитные мониторы дают гораздо худшее качество изображения и в настоящее время с компьютерами не используются. Однако остается опасность купить давно выпущенный экземпляр, соблазнившись "бросовой" ценой. Как RGB, так и композитные мониторы могут быть только цветными, но монохромные мониторы совместимы с RGB по разъемам.

Различие между аналоговыми и цифровыми мониторами во многом совпадает с различием между композитным и RGB-мониторами, но судьба их противоположна: цифровые вытесняются аналоговыми.

Монохромный аналоговый монитор получает сигнал по одной паре проводов (каналу); для управления цветным (RGB) аналоговым монитором нужны три канала (а значит, четыре провода: по одному на каждый основной цвет и общая земля). Яркость каждой точки экрана строго пропорциональна напряжению управляющего сигнала. Такая пропорциональность обеспечивается дорогостоящей высококачественной электроникой, однако она позволяет плавно изменять яркость и цвет любой точки экрана. Число уровней яркости и оттенков определяет адаптер; монитор их не ограничивает.

Цифровые мониторы, напротив, обеспечивают вывод лишь ограниченного числа уровней яркости и цветов. Они позволяют включать/выключать по одному каналу только один уровень яркости. Управление несколькими уровнями интенсивности приходится разделять по разным проводам, как цвета в RGB-мониторах.

Для простоты рассмотрим формирование изображения на монохромном цифровом мониторе. При использовании одного канала пиксел может быть либо выключен (черный цвет), либо может светиться с фиксированной интенсивностью. Итак, яркость имеет два уровня: "нет сигнала" и "есть сигнал". Две линии позволяют формировать уже четыре уровня яркости: черный цвет соответствует отсутствию сигнала по обоим каналам, единичная яркость - сигналу по первому каналу, двойная - сигналу по второму и тройная - по обоим одновременно.

Примером может служить очень неплохой монитор, который в свое время выпускала фирма IBM. Официально он называется IBM Monochrome Display, но чаще употребляется название TTL Monochrome Monitor (монохромный монитор TTL). Сокращение TTL означает, что этот монитор цифровой. Он обеспечивает четыре уровня яркости; впрочем, большинство адаптеров используют только три - черный, стандартный и яркий (глаз плохо различает два ярких уровня).

При формировании изображения на цветных мониторах те же рассуждения надо применить к каждому из трех основных цветов. Поэтому для передачи, скажем, 16 цветов цифровой монитор требует 4 канала, а для 64 цветов нужно 6 линий.

Приведем еще два примера. Монитор Color Graphics Monitor фирмы IBM, чаще называемый просто CGA-монитором, получает информацию по четырем линиям. Три из них включают/выключают основные цвета, а сигнал по четвертому каналу увеличивает яркость сразу всех цветов. Такая система управления называется RGB; буква I обозначает интенсивность.

Второй монитор - IBM Enhanced Color Display (ECD); обычно его называют EGA-монитором. Он допускает вывод 64 цветов, и для этого принимает сигнал по шести каналам, обозначаемых буквами RGRgb. Линии rgb управляют 80-процентным уровнем интенсивности каждого из основных цветов.

Пока адаптеры позволяли выводить всего от 2 до 16 цветов, не стоило тратить средства на монитор, который способен точно отобразить все промежуточные уровни яркости. В настоящее время положение изменилось. Конечно, управление большим числом уровней целесообразнее объединять в один сигнал не в мониторе, а в самом адаптере - хотя бы для того, чтобы не тянуть между ними толстый кабель, состоящий из десятков проводов. Поэтому до 1987 г. использовались почти исключительно цифровые мониторы, теперь же их вытесняют аналоговые.

Так, концерн IBM начал в 1987 г. выпуск адаптера VGA и предназначенных для работы с ним аналоговых мониторов (как цветных, так и монохромных). Вскоре он полностью прекратил выпуск мониторов MA, CGA и EGA. Однако они все еще широко распространены, а совместимые с ними устройства продолжают изготавливаться многими фирмами.

Вообще уместно заметить, что мониторы, взаимозаменяемые с изделиями IBM, появляются на рынке мгновенно. Похоже, что их изготовители не испытывают затруднений технического характера.

Поскольку устройство ПК позволяет легко заменить адаптер, а качество видеоподсистем постоянно улучшается, на Западе многие пользователи модернизируют свои машины. Новый адаптер стоит недорого, но цветной монитор обходится в 2-3 раза дороже адаптера. Известная японская фирма NEC несколько лет назад выпустила монитор "Мультисинк", способный автоматически синхронизироваться с различными адаптерами, меняя для этого в широких пределах частоту кадров, строк и сигналов, а значит, и разрешение (от 640 x 200 до 800 x 600 по паспорту, а фактически до 800 x 600). Кроме того, этот монитор можно переключать из цифрового режима работы в ана-

логовый и обратно. Примерно в то же время фирма IBM, а за ней и другие компании, стали выпускать адаптеры второго поколения. Для работы с ними многие пользователи предусмотрительно приобрели мониторы типа мультисинк. Дальновидность такого решения подтвердилась при появлении адаптеров третьего поколения (аналоговых).

Естественно, многочастотные мониторы стали выпускать и другие фирмы. Они не могут называть их "Мультисинк", поскольку это название является торговой маркой концерна NEC и охраняется законом. Однако пользователи все равно называют такие мониторы мультисинками.

Помимо "совместимости с будущим", многочастотные мониторы обладают еще одним важным достоинством: они резко удешевляют нестандартные подсистемы видео. Дело в том, что разработка и подготовка производства адаптеров требуют намного меньших расходов, чем налаживание производства мониторов. Теперь изготовители адаптеров могут рассчитывать на мультисинки.

Заметим, что стандартные мониторы тоже могут менять частоту синхронизации, но не в широких пределах, а между несколькими фиксированными уровнями. Так, монитор EGA для эмуляции CGA снижает частоту строк и частоту видеосигнала, а VGA меняет частоту видеосигнала и частоту кадров не только для эмуляции EGA, но и при переходе из текстового режима в графический.

Приведем таблицу основных характеристик наиболее распространенных мониторов:

	TTL	CGA	EGA	VGA	Мульти- синк
Частота кадра (Гц)	50	60	60	70	50-70
Частота строк (КГц)	18,432	15,750	21,810	31,500	16-36
Полоса (МГц)	16,257	14,318	16,257	28,322	39 и выше
Разрешение	720x350	640x200	640x350	640x480	800x600

Многочастотные мониторы образуют большое семейство и обладают различными характеристиками; приведенные в таблице относятся к монитору "Мультисинк" фирмы NEC.

Мониторы одного класса, обладающие одинаковыми принципиальными характеристиками, различаются конструкцией. Среди наиболее важных параметров назовем размер экрана, его форму (точнее, пропорции), цвет фосфора монохромных мониторов. Есть и другие конструктивные особенности, о которых будет говориться ниже. Многие характеристики понятны без пояснений (вес, дизайн, расположение ручек управления) и мы не станем их обсуждать.

Монохромные мониторы выпускаются с кинескопом, покрытым зеленым, желтым ("янтарным") и белым фосфором. Желтые мониторы предпочтительны для работы в светлом помещении, а зеленые - в затемненном. Белые монохромные мониторы появились сравнительно недавно. Они особо хороши для ввода и редактирования текстов, в настольных издательских системах и вообще при имитации работы с бумагой. По-английски их так и называют: paper-white, т.е. белые, как бумага.

На монохромных мониторах обычно выводится позитивное изображение - светлые буквы и линии на темном фоне. Инверсное или негативное изображение - когда тот же видеосигнал соответствует черным символам на светлом фоне - применяется в ос-



новном в мониторах с белым фосфором. Некоторые мониторы снабжены переключателем инверсии. Конечно, того же эффекта можно добиться и программным путем, но большинство пользователей работает с готовыми программами, а в них не всегда предусмотрено управление цветом.

Экран монитора имеет размер от 9 до 24 дюймов (22 - 61 см) по диагонали. Мониторы больших размеров не только дорого стоят: если они расположены на столе в 50-70 см от пользователя, то в его поле зрения помещается только часть экрана. Такие мониторы применяются в основном в системах автоматизированного проектирования, настольных издательских системах и для других графических работ, когда внимание сосредоточено на небольшой части кадра, а также в тех случаях, когда на экран смотрят издалека.

Небольшие экраны, напротив, слишком мелки и утомляют зрение: они применяются в переносных машинах и там, где для монитора нормального размера не хватает места. Для большинства работ оптимальный размер экрана равен 12-14 дюймам (30-35 см). При этом на небольших экранах получается более четкое изображение, а крупные буквы на экранах большего размера меньше утомляют глаз.

У подавляющего числа мониторов горизонтальная сторона экрана относится к вертикальной в пропорции 4:3. Иногда встречаются мониторы с вертикальной ("портретной") ориентацией сторон и еще реже - квадратные. Мониторы с портретной ориентацией используются в основном в настольных издательских системах, где позволяют имитировать целую страницу текста. Впрочем, мониторы большого размера, которые дают возможность рассматривать изображение сразу двух страниц ("разворот"), в этом отношении еще удобнее.

Следует отметить, что размеры и пропорции экрана не связаны непосредственно с его разрешением. Пиксели могут находиться на разном расстоянии друг от друга как по вертикали, так и по горизонтали. Если мониторы с разрешением 640 x 200, 640 x 480 и 1024 x 768 выпускаются в одинаковом корпусе, то в выключенном состоянии они неразличимы. Однако определенное соответствие между разрешением и геометрическими размерами все же имеется: небольшой экран с высоким разрешением трудно изготовить, а на большом экране с низким разрешением точки зрителью не сливаются в единое изображение. Вертикальное разрешение мониторов с портретной ориентацией больше горизонтального.

Отражение от экрана посторонних источников света, особенно осветительных ламп, мешает работе и портит зрение. Чтобы ослабить такие блики, экран многих мониторов либо изготавливают из матового стекла, либо наносят на него матовое покрытие.

В последнее время появились так называемые мониторы с плоским экраном. Фактически они все же выпуклые, но намного меньше, чем обычные. Такие мониторы немного дороже стандартных, поскольку чуть сложнее по конструкции (требуется переменная фокусировка электронного луча). Как правило, это белые мониторы, идеальные для работы с текстами. Кстати, одна фирма выпускает цветной монитор с плоским экраном (очень дорогой). Монитор с плоским экраном легче повернуть так, чтобы от него не отражались лампы.

Персональный компьютер обладает огромными возможностями наглядного представления информации. Данные и различные объекты можно показать в цвете и движении. Умело пользуясь этими возможностями компьютера, вы сможете создавать и оживлять графические объекты, манипулировать цветом.



© Михаил Сальников

## ...и адаптеры

Но прежде всего вы должны разобраться в аппаратных средствах. Возможности визуального представления информации на персональном компьютере определяются типом дисплея и видеоадаптера, к которому он подсоединен. Обычно стараются подобрать дисплей, на котором полностью можно использовать возможности видеоадаптера.

Видеоадаптер, как правило, отдельная плата, на которой расположены микросхемы оперативной памяти самого адаптера, контроллер дисплея и микросхемы с программным обеспечением.

Адаптеры могут работать в одном из алфавитно-цифровых (текстовых) или графических режимов. В алфавитно-цифровом режиме на экране (точнее, в специальном видеобуфере компьютера) адресуется каждый элемент в фиксированном (в зависимости от режима) числе столбцов и строк экрана. Обычно экран содержит 80 столбцов и 25 строк, хотя возможны алфавитно-цифровые режимы 80 x 43 и 40 x 25. В графическом режиме адресуется каждая отображаемая точка экрана (pixel). Именно поэтому в графике можно указать цвет каждой точки экрана и нарисовать любую геометрическую фигуру или создать целую картину. В зависимости от режима работы адаптера и типа дисплея, экран может быть разбит, например, на 350 x 200 или 640 x 350 точек. Ясно, что для адресации такого поля в графических режимах потребуется гораздо больший объем памяти, чем в алфавитно-цифровых. Обычно при включении компьютера и загрузке операционной системы адаптер находится в алфавитно-цифровом режиме. Можно ли, глядя на дисплей, определить режим? Да, можно: в графических режимах, если это специально не инициируется активной в данный момент программой, курсор не отображается, а указатель "мыши" имеет форму стрелочки (в текстовом режиме указатель "мыши" - прямоугольник).

Первые видеоадаптеры (1981 г.), разработанные для персональных компьютеров, - MDA (Monochrome Display Adapter) и CGA (Color Graphic Adapter). MDA поддерживает только алфавитно-цифровой (80 x 25) режим работы, CGA - алфавитно-цифровые (80 x 25 и 40 x 25) и графический (в зависимости от типа монитора максимальное разрешение 640 x 200 точек). CGA допускает работу с четырьмя основными цветами.

Затем был создан адаптер HGC (Hercules Graphics Card, 1982 г.) и позднее - HGC+ (Hercules Graphics Card Plus, 1986 г.). Оба адаптера могут работать только в монохромном графическом режиме с различными оттенками. HGC+ в отличие от HGC может использовать не только хранящиеся в постоянной памяти адаптера символы, но и специальные, загружаемые в оперативную память адаптера шрифты. Основная особенность этих графических карт в том, что они работают с мониторами высокого разрешения и позволяют работать с более эргономичными шрифтами, так как на каждую букву шрифта приходится, по сравнению с другими адаптерами, больше точек.

В 1985 г. появился адаптер EGA (Enhanced Graphics Adapter). Он допускает работу как в текстовом, так и графическом режимах. Внутренняя оперативная память адаптера составляет 256 Кбайт. В графическом

режиме стандартное максимальное разрешение - 640 x 350 точек (некоторые модификации EGA допускают разрешение 640 x 480 точек).

Цветовые палитры адаптера позволяют пользоваться 64 цветами, но одновременно работать можно только с 16. Доступ к остальным цветам осуществляется переходом к другой палитре.

В серии персональных компьютеров PS/2 (1988 г.) используется новая графическая система VGA (Video Graphics Array), представляющая собой часть аппаратных средств видеоподсистемы компьютеров этой серии. VGA - это одна микросхема, интегрирующая в себе совокупность микросхем адаптера EGA. VGA также допускает работу в алфавитно-цифровых и графических режимах. Максимальное разрешение в графическом режиме - 640 x 480 точек. Поддерживается одновременная работа с 256 цветами из 262144. Принципиальное отличие VGA от других адаптеров в том, что он генерирует аналоговые, а не цифровые сигналы управления цветом. Именно это дает возможность увеличить число цветов. С компьютерами предыдущей серии (PC/AT) VGA работает как отдельный графический адаптер.

Существует еще ряд адаптеров, например MCGA (Multi-Color Graphics Array), часть видеоподсистемы младших моделей серии PS/2, Hercules InColor Card, Hercules Color Card. Кроме того, адаптеры одного типа могут иметь различные параметры (так, EGA может иметь не 256, а 64 Кбайта внутренней оперативной памяти. В этом случае можно работать только с 4 цветами. Позднейшие модификации адаптера EGA допускают максимальное разрешение 640 x 480 точек).

Мы не будем здесь подробно описывать все существующие графические адаптеры. Рассмотрим лишь один - EGA. Выбор пал на него не случайно. Сегодня это самый распространенный цветной адаптер, позволяющий создавать полноценные графические приложения. С одной стороны, EGA способен эмулировать режимы CGA (а некоторые модификации - и режим Hercules), с другой - его программный интерфейс совместим с программным интерфейсом VGA. Совместимость означает, что большинство программ, написанных для EGA, без изменений пойдут и на VGA.

Адаптер EGA может работать в различных режимах, которые нумеруются, начиная с 0.

Номер	Разрешение	Число цветов	Тип режима
0	40 x 25	16	Текстовый
1	40 x 25	16	Текстовый
2	80 x 25	16	Текстовый
3	80 x 25	16	Текстовый
4	320 x 200	4	Графический
5	320 x 200	4	Графический
6	640 x 200	2	Графический
7	80 x 25	2	Текстовый
13	320 x 200	16	Графический
14	640 x 200	16	Графический
15	640 x 350	2	Графический
16	640 x 350	16	Графический





Вы, наверное, уже заметили, что параметры некоторых режимов совпадают, а нумерация режимов не последовательна (в ней есть пропуски). Режимы с 8 по 10 разработаны для промежуточной серии персональных компьютеров (PCjr), а режимы 11 и 12 используются внутренним программным обеспечением EGA. Для адаптера VGA возможны еще режимы 17, 18 и 19. Совпадение же параметров различных режимов для EGA объясняется тем, что нумерация режимов унифицирована для разных типов адаптеров персонального компьютера. После включения компьютер с EGA находится в 3 режиме.

Каким образом можно установить нужный режим? Как определить, в каком режиме находится ваша видеосистема? В системном программном обеспечении, вложенном в одну из микросхем компьютера, содержится комплекс программ базовой системы ввода/вывода - BIOS (Basic Input/Output System). BIOS осуществляет связь программ со всеми внешними устройствами компьютера. Если в программе на языке высокого уровня вы записываете предложение ввода/вывода, компилятор заменяет его на обращение к соответствующей программе BIOS, передавая ей необходимые параметры. Строго говоря, какие программы BIOS расположены в чипе, а какие в файле операционной системы, зависит от того, какой системой вы пользуетесь - MS-DOS или PC DOS.

Для программиста связь с BIOS осуществляется в обеих системах одинаково - с помощью прерываний. Прерывание (в нашем контексте) представляет собой специальный механизм обращения к программам системного матобеспечения. Все прерывания имеют номера. Для видеосистемы отведено прерывание с номером 16 или в шестнадцатеричном представлении 10h. В рамках этого прерывания можно запросить выполнение многих функций. У каждой функции есть свой номер. Получить номер текущего видеорежима можно с помощью функции 15 (0Fh), а установить режим - с помощью функции 0 (00h). Номер функции должен быть передан системе в одном из регистров процессора. После выполнения функции система передает информацию программе также в одном из регистров процессора. Номер функции всегда передается в регистре AH. Функция 0Fh возвращает номер текущего режима в регистре AL (на самом деле AH и AL - "старший" и "младший" байт регистра AX). Для того чтобы успешно работать, вам нужно ознакомиться с архитектурой микропроцессора, на котором построен ваш компьютер!.

Для дальнейшей работы нам понадобится "общий язык". В качестве такого языка, на наш взгляд, больше всего подойдет Си - гибкий и мощный язык программирования высокого уровня. Мы будем использовать два диалекта этого языка: диалект, реализованный в компиляторе Turbo C 2.0 фирмы BORLAND и диалект, реализованный в Microsoft 5.1 фирмы MICROSOFT (или Quick C 2.0). Для каждого примера программы или фрагмента компилятор будет точно указан. Отладка всех программ будет выполняться в MS-DOS 3.3.

Установить режим можно так (Turbo C):

```
#include<dos.h>
setmode(mode)
/* подпрограмма установки видеорежима */
int mode; /* mode - номер режима */
{
union REGS regs;

regs.h.al = mode;
regs.h.ah = 0; /* 0 - номер функции */
int86(0x10,&regs,&regs); /* вызов прерывания */
}
```

Получить номер текущего режима можно с помощью такой подпрограммы:

```
#include<dos.h>
getmode()
/* подпрограмма получения номера текущего режима */
{
union REGS regs;

regs.h.al = 0;
regs.h.ah = 0x0f; /* номер функции */
int86(0x10,&regs,&regs); /* вызов прерывания */
return(regs.h.al); /* возвращение номера режима */
}
```

Позэкспериментируйте с этими подпрограммами на компьютере. Напишите основную программу, параметром вызова которой будет номер устанавливаемого режима или знак вопроса, означающий вывод на экран номера текущего режима.

Создание графических образов и их анимация ("оживление") - наиболее привлекательная сторона работы с персональным компьютером, но это требует от пользователя определенного профессионализма, который невозможен без ясного представления о работе и взаимосвязи всех компонент графического адаптера. С одной стороны, такие знания окажут вам неоценимую помощь в процессе отладки программных средств и дадут возможность принять оптимальные технические решения. С другой стороны, работа только с функциями BIOS не позволяет создавать полноценную сложную графику в движении. Для этого необходимо научиться работать непосредственно с программируемыми частями аппаратуры, и здесь иногда будет нужен язык Ассемблера (мы будем пользоваться Microsoft Macro Assembler 5.1).

Для нас в видеосистеме персонального компьютера будут представлять интерес следующие компоненты, доступные программы:

- \* контроллер дисплея;
- \* регистры состояния;
- \* управляющие регистры;
- \* видеобуфер.

Контроллер дисплея графического адаптера EGA выполнен на разработанных фирмой IBM больших интегральных схемах (БИС), содержащих набор управляющих регистров, каждый из которых отвечает за определенную функцию (начало и конец горизонтального мерцания, начало и конец курсора и т.д.). Контроллер выполняет вертикальную и горизонтальную синхронизацию, обеспечивая отображение на дисплее содержимого видеобуфера в нужном цвете и нужной яркости (EGA допускает одновременный вывод на экран двух несмежных областей видеобуфера). Именно этот контроллер наращивает указатель адреса видеобуфера, который затем применяется схемами дисплея для выбора и отображения данных. Программист может обратиться к регистрам контроллера, используя соответствующий адрес порта.

В качестве порта ввода/вывода программисту доступен 8-битовый регистр состояния. Адрес порта для EGA в конфигурации с цветным дисплеем - 3DAh. Регистр содержит биты состояния, которые можно использовать для синхронизации вывода изображения с циклом обновления экрана, чтобы уменьшить эффект "смогивания" при смене картинки. В некоторых модификациях EGA имеется несколько регистров состояния. В любом случае эти регистры доступны только для чтения.

Управляющие регистры дают возможность в дополнение к контроллеру дисплея выполнять различные

функции. Например, в зависимости от режима работы адаптера (текстовый, графический) подключать или отключать генератор символов, устанавливать размеры символов в пикселях и т.д. В EGA все эти регистры доступны только для записи, прочесть их нельзя. Регистры управления режимом делятся на несколько групп. К каждой группе можно адресоваться как кциальному порту ввода/вывода. Первая группа регистров (Sequencer) содержит пять регистров данных. Доступ к регистрам осуществляется через 2 порта - 3C4h и 3C5h, первый из которых служит для адресации и задания номера выбираемого регистра (0 - 4). Второйпорт используется для пересылки данных в выбранный регистр. Рассматриваемая группа регистров генерирует внутренние сигналы синхронизации при адресации видеопамяти. Кроме того, регистры этой группы позволяют защитить от записи нужную часть видеобуфера.

Вторая группа регистров (Graphics Controller) отвечает за передачу данных между видеобуфером, центральным процессором и третьей группой регистров (Attribute Controller). Во вторую группу входит 9 регистров данных, доступ к которым осуществляется через порты 3CEh (адресация) и 3CFh (данные). Третья группа регистров используется для работы с цветом (16 цветовыми палитрами). 20 регистров данных этой группы и ее регистр адреса закреплены за одним портом 3C0h. Значение, посыпаемое напорт, заносится в регистр адреса или данных в зависимости от внутреннего переключателя этой группы. Для того чтобы установить переключатель в EGA, нужно прочесть регистр состояния дисплея (порт 3DAh).

Видеобуфер - это область памяти в видеосистеме, в которой хранятся отображаемые на экране данные. Полная память EGA обычно составляет 256 Кбайт, разделенные на 4 блока по 64 Кбайта. Блоки параллельны, т.е. в полном адресном пространстве центрального процессора они занимают одни и те же адреса памяти. Управляющие регистры адаптера позволяют осуществлять доступ к блокам памяти последовательно или параллельно. Каждый пиксель экрана кодируется одним битом в каждом из блоков видеобуфера. Таким образом, каждый байт с уникальным адресом видеобуфера содержит информацию о 8 пикселях экрана. Например, предположим, что байты с одним и тем же адресом в различных блоках имеют следующую конфигурацию бит:

Блок 0	1 0	0 1	0 0	0 1
Блок 1	0 1	0 0	0 1	1 0
Блок 2	0 0	0 1	0 1	1 1
Блок 3	0 1	0 1	0 1	0 0

Первый пиксель будет закодирован конфигурацией 0001, второй - 1010 и т.д. Конфигурация получается конкатенацией одних и тех же бит из различных блоков. Полученное сочетание еще не является идентификатором цвета пикселя на экране. Цвет пикселя получается логическим умножением этой кодировки на содержимое одного из управляющих регистров (Color Plane Enable register), что позволяет использовать 64 различных цвета. Результат умножения (4 бита) служит номером одного из 16 регистров палитры, входящих в управляющие регистры адаптера. Выбранный регистр палитры и определяет цвет отображаемого пикселя. Стандартное содержимое регистров палитры формируется BIOS при установке графического режима. Важно понять, что в полном режиме EGA обмен информацией между видеобуфером и экраном не может осуществляться простым копированием информации, как это делается в CGA. В EGA при передаче информации обязательно используются управляющие регистры.

В действительности каждая операция обмена между центральным процессором и видеобуфером вызывает передачу 4 байт информации (по одному для каждого блока видеопамяти). С каждым блоком памяти связан один 8-битовый служебный регистр. Обмен данными идет только через эти регистры. Управление обменом данными между центральным процессором, видеобуфером и служебными регистрами возложено на вторую группу регистров управления режимом (Graphics Controller). Реально в центральный процессор передается 1 байт информации. Способ его формирования задается в одном из регистров этой группы (Graphics Mode register). Центральный процессор может принять байт из отдельного блока видеопамяти или 4 бита, сформированных конкатенацией бит из разных блоков. В процессе обмена возможны битовые логические операции над информацией в служебных регистрах.

Поскольку в EGA каждый пиксель кодируется одним битом в каждом из четырех блоков видеопамяти, полной памяти EGA достаточно для двух графических страниц при графическом режиме с максимальным разрешением. Видеобуфер последовательно заполняется информацией о каждом пикселе, начиная с верхнего левого угла экрана и далее построчно.

На первый взгляд вам может показаться, что для прикладного программирования все это слишком сложно. На самом деле это не так. В следующих статьях мы приведем более подробные сведения о возможностях видеoadаптера EGA и приемах работы с ним, а также рассмотрим процесс создания и анимации графических объектов, работу с цветом.



# Карта телефонного модема

Обмен информацией - основная деятельность человека. С развитием вычислительной техники для сбора, хранения и передачи информации стали использовать компьютеры. Идея передачи информации между компьютерами привела к мысли объединить их в локальные сети. Чтобы осуществить такую связь, применяются сетевые адAPTERы.

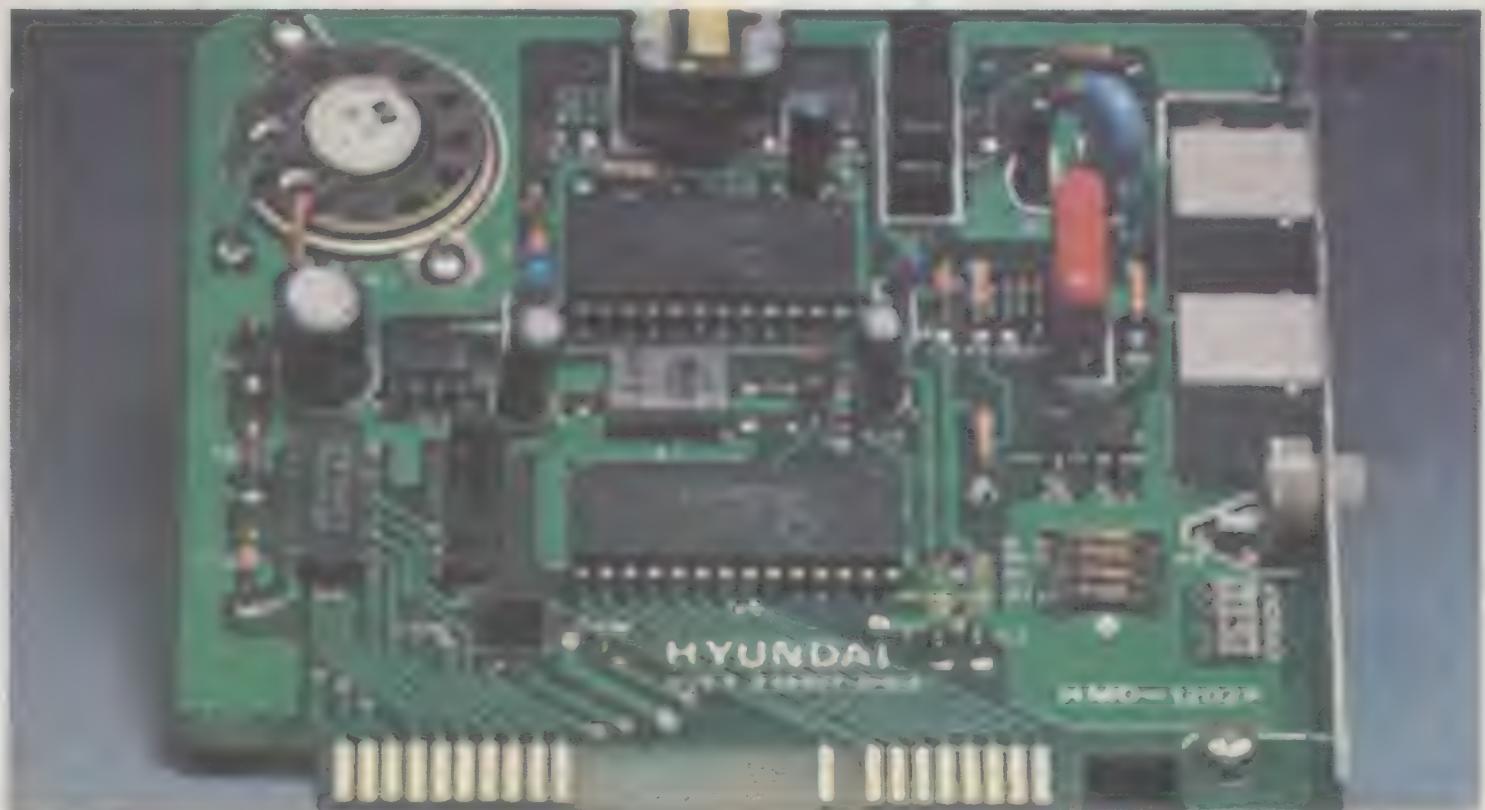
Таким образом можно получить быстродействующую и надежную сеть, работающую, однако, на довольно ограниченной территории (в пределах одного отделения, предприятия), поскольку расстояние между соединенными компьютерами, как правило, не превышает 1000 м.

Другим методом соединения компьютеров является модемная связь, в основе которой лежит существующая телефонная сеть. Связь такого рода особенно удобна в случае передачи информации на большие расстояния, а также для непрофессиональных пользователей компьютеров. Она позволяет им обмениваться сообщениями, дает возможность пользоваться общедоступными банками информации, организовать любительские компьютерные сети в международном масштабе. Дальность такой сети практически не ограничена и зависит только от состояния и качества телефонной сети.

Конструкторы адAPTERов, являющихся расширениями компьютеров стандарта ПК, разработали несколько разновидностей карт, которые обеспечивают подключение компьютера к телефонной линии. Карта модема - это устройство, с помощью которого цифровая информация на выходе из компьютера преобразуется в модулированный акустический сигнал. В действительности он является электрическим сигналом акустической частоты, направляемым на телефонную линию посредством гальванического соединения. Когда компьютер применяется для приема информации из телефонной сети, модем должен принять сигнал из телефона и преобразовать его в цифровую информацию, "понятную" компьютеру. На выходе модема информация подвергается Модуляции, а на входе - ДЕМодуляции. Отсюда и название МОДЕМ.

Стандартная телефонная линия приспособлена к передаче акустических сигналов частотой от 300 до 3400 Гц и полностью обеспечивает передачу речевой информации. Параметры линии недостаточны для цифровой передачи данных непосредственно между компьютерами, поскольку при таком виде трансмиссии сигнал намного превышает возможности телефонной сети. Назначение модема заключается в замене сигнала, поступающего из компьютера (сочетание нулей и единиц), электрическим сигналом с частотой, соответствующей рабочему диапазону телефонной линии. Акустический канал этой линии модем разделяет на две полосы низкой и высокой частоты. Полоса низкой частоты применяется для передачи данных, а полоса высокой частоты - для приема данных. В выпускаемых в настоящее время картах телефонных модемов используются два способа кодировки информации: для медленных модемов - передача со скоростью до 300 бод (бит/с) и для быстрых модемов - до 2400 бод.

В медленных модемах метод кодировки назван FSK (Frequency Shift Keying) и заключается в использовании четырех выделенных частот сигнала. При передаче информации сигнал частотой 1070 Гц обозначает логический нуль, а сигнал частотой 1270 Гц - логическую единицу. При приеме данных сигнал в 2025 Гц - это логический нуль, а сигнал в 2225 Гц соответствует логической единице.



Если же скорость превышает 300 бод, применяется второй метод кодировки сигнала телефонной линии, называемый PSK (Phase Shift Keying). В этой системе используются две частоты сигнала. Сигнал частотой 2400 Гц предназначен для передачи данных, а сигнал частотой 1200 Гц - для приема информации. Метод PSK состоит в передаче двух битов информации одновременно. Существуют четыре сочетания нулей и единиц: 00, 01, 10, 11. Любая из этих возможностей представлена сигналом постоянной частоты, но с другой фазой: сочетание битов 00 - это сигнал со сдвигом 0°, 01 - сдвиг 90°, 10 - сдвиг 180°, 11 - сдвиг 270°.

Карта телефонного модема, кроме кодировки сигнала, выполняет основные функции модема. К ним следует отнести:

- \* обнаружение сигнала вызова собственного номера;
- \* передачу на телефонную линию сигнала набора номера другого абонента;
- \* обнаружение сигнала вызова номера другого абонента;
- \* обнаружение сигнала занятости номера другого абонента;
- \* передачу сигнала ответа на вызов другим абонентом;
- \* определение характера вызова - передача данных или обычный разговор.

Контроллер модема - это, как правило, специализированный микрокомпьютер типа SC 11007 или SC 11008, содержащий 8-разрядное АЛУ, ПЗУ 8 Кбайт, ОЗУ 128 байт, таймер, командный регистр, контроллер прерываний, стек, порт ввода/вывода. Если карта модема присоединена к системной шине компьютера ПК и взаимодействует с шиной адресов и данных, то применяется "параллельный" контроллер SC 11007. Если же карта работает с компьютером посредством канала связи RS 232, то используется "последовательный" контроллер SC 11008. В некоторых конструкциях роль контроллера выполняет процессор 8031 с внешним ПЗУ (2732, 2764) и схемой 74LS373.

Модем - электронная схема, обнаруживающая, кодирующая и декодирующая сигнал телефонной линии; как правило, это схема типа SC 11004 с квадратным резонатором. Усилитель с громкоговорителем выполняет ту же роль, что и звонок телефона.

Система переключения "телефон-компьютер" также может управляться пользователем.

Установка карты модема заключается в ее подключении к разъему расширения на базовой плате ПК и подсоединении к гнездам на карте модема проводов телефонной линии и телефонного аппарата. После загрузки коммуникационной программы модем готов к работе. Коммуникационная программа принимает и записывает на дискету информацию из модема, а также передает в модем данные, указанные пользователем. Многие коммуникационные программы (XTALK, COMM, MODEM8, XCOM и т.п.) позволяют применять модемную связь не только для простых соединений типа "отправь/прими", но и для определения очереди файлов при передаче информации в запрограммированное время, а также для диалогового режима связи и т. д.

Наличие модемной карты в компьютере дает возможность доступа к сети модемной связи. Информацию по этой теме читатели найдут на страницах нашего сборника в статье о сети FIDO.

Перевод Романа Хмельевского

С
\компьютер на работе\
С

© Казимеж Цянцяра

# Сети

# NOVELL

Компьютерные сети, и прежде всего так называемые локальные вычислительные сети, пользуются огромной популярностью во всем мире. Что послужило причиной столь бурного развития ЛВС и каковы перспективы их развития?

ЛВС - естественное следствие появления микрокомпьютеров. Не только на больших комбинатах, но и на маленьких предприятиях они могут быть неоценимыми помощниками во всех случаях, когда микроЭВМ используются для облегчения и ускорения труда человека. В качестве примера приведем следующие их достоинства.

1. Экономия места (дисковые ЗУ). Как показывает практика, около 50% программных продуктов, установленных в ЭВМ в учреждениях, на предприятиях или в проектных бюро, - это одни и те же программы. Применение ЛВС позволяет работать сразу нескольким пользователям с удаленными массивами данных, находящимися в избранном компьютере сети (file server). Такое решение дает ощутимую экономию дисковой памяти периферийных компьютеров.

2. Сложная система защиты, при которой вероятность ошибок, связанных с записью информации на жесткий диск, чрезвычайно мала (версия SFT - System Fault Tolerant).

3. Локальная вычислительная сеть обеспечивает свободную связь между отдельными пользователями. Благодаря применению "компьютерной почты" появилась возможность передачи документов абонентам, находящимся на большом расстоянии друг от друга.

4. Благодаря соответствующим программным средствам можно использовать работающие в сети микроЭВМ в качестве терминалов, соединять ЛВС друг с другом или с глобальными сетями (ГВС).

По данным, представленным на международной ярмарке "СеBIT '89" в Ганновере, в 1988 г. в Европе было установлено около 33 000 локальных сетей. Эксперты предусматривают, что в 1992 г. число действующих ЛВС в Западной Европе превысит 370 000.

Файл-серверами, в которых устанавливается операционная система NetWare фирмы NOVELL, являются, как правило, 16-разрядные компьютеры, совместимые с IBM PC и PS/2, с микропроцессором Intel-80286 или Intel-80386. Серверы предназначены для хранения большинства используемых файлов. Существует также версия NetWare VMS, где файл-сервером может являться мини-ЭВМ DEC VAX, работающая под контролем ОС VMS 4.6 или 4.7. Остальные узлы могут быть микроЭВМ, совместимыми с IBM PC/XT, AT или PS/2.

Рабочими станциями, действующими в сетях фирмы NOVELL, могут быть микроЭВМ IBM PC/XT/AT/386, IBM





PS/2 (работающие под контролем ОС DOS 2.x, 3.x, 4.x, OS/2, PC-MOS или Windows-286/386) и микроЭВМ типа Macintosh. Есть возможность применения рабочих станций, действующих под контролем ОС UNIX.

Версии ELS (Entry Level Solution) и Advanced NetWare могут совмещать функции файл-сервера с функцией рабочей станции (это касается, как правило, малых сетей), но это не рекомендуется по соображениям безопасности. Сбой программы, запущенной на сервере, однозначен блокаде доступа к находящимся в нем данным.

Система NetWare не зависит от технологии сети и применяемых сетевых адаптеров. Одна из ее последних версий - 2.15 - допускает применение большинства доступных на рынке сетевых адаптеров: 3Com Etherlink, Standard Microsystems Arcnet, IBM PC Token Ring, IBM PC Network, D-Link, Ethernet и пр., а также разных типов кабеля: витой пары, коаксиальных, световодных. Для соединения ЛВС или подключения удаленных рабочих станций возможно также применение интерфейса RS-232.

Фирма NOVELL является одним из крупнейших производителей операционных систем для локальных сетей, составленных из микроЭВМ класса IBM PC. По материалам DataQuest, NOVELL контролирует большую часть мирового рынка ЛВС. Фирма предлагает несколько версий системы NetWare, предназначенных для различных групп пользователей и разного оборудования:

- \* ELS NetWare Level I - до 4 пользователей;
- \* ELS NetWare Level II - до 8 пользователей;
- \* Advanced NetWare - до 100 пользователей;
- \* SFT NetWare - до 100 пользователей.

Самая развитая ОС фирмы NOVELL - SFT NetWare. Она обеспечивает обслуживание сети с большим количеством пользователей и устраниет возможные ошибки при записи информации на диск файл-сервера. Для такой ОС требуется компьютерное оборудование с дублированными контроллерами и жесткими дисками, оснащенное дополнительно устройствами поддержки питания UPS (Uninterrupted Power Supply).

Система NetWare обслуживает сеть с файл-серверами, предоставляющими остальным узлам свои аппаратные ресурсы: диски и принтеры. В каждой сети может работать несколько файл-серверов.

Кроме файл-серверов, можно выделить специализированные узлы сети для обслуживания каналов связи, а также другие, являющиеся "мостами" (bridges), соединяющими несколько ЛВС в одну ГВС. Кроме того, существует возможность подключить специализированные узлы (value-added servers), "ответственные" за распечатку, архивацию и обработку данных (print servers, archive servers, batch job servers и database servers). ОС NetWare была разработана специально для сетевой среды и гораздо эффективнее по сравнению с сетевыми решениями, основанными на системе DOS: IBM PC Network и пр.

В ЭВМ, выполняющей роль файл-сервера, можно одновременно установить до 4 сетевых адаптеров (также разных стандартов) и до 4 дисковых каналов, обслуживающих дисковые ЗУ общей емкостью до 2 Гбайт, а также ОЗУ до 12 Мбайт. В случае крупной сети и дисков большой емкости, объем ОЗУ существенно влияет на скорость обслуживания дисковых операций. Сетевые платы и дисковые каналы обслуживаются параллельно без посредства BIOS в режиме Protected Mode микропроцессора 80286. Файл-сервер может обслуживать также до 5 принтеров, подключенных к портам стандарта Centronics (до 3) и COM 1/COM 2.

Дополнительно фирма NOVELL разработала эмулятор протокола NetBIOS, что позволяет использовать программные средства, предназначенные в сетевой версии для IBM PC Network.

Важным достоинством системы NetWare является ее высокая отказоустойчивость, достигнутая благодаря ряду решений, которые до сих пор применялись только в больших ЭВМ. Они состоят прежде всего в дублировании ТРФ и самих файлов, а также в текущем контроле за состоянием сети.

Система NetWare обеспечивает возможность присоединения локальной сети к большим ЭВМ (HOST) и АРМ типа IBM, DEC, SUN, APOLO и другим путем применения сетевой платы стандарта Ethernet MICO/MICOM/Interlan NP600A и протокола TCP/IP, позволяющего пользователям сети работать в режиме эмуляции терминала (Telnet) или передачи файлов (FTP). Для соединения системы NetWare с большими системами ВМ (например, IBM-4341, IBM-370), предназначен целый ряд программных пакетов PCOX, позволяющих пользователям сети работать в режиме эмуляции терминалов IBM-3270 или IBM-5250.

Существует множество пакетов прикладных программ, предназначенных для пользователей ЛВС. Следует отметить, что на любом из работающих в сети компьютеров могут быть использованы программы для персональных ЭВМ класса IBM PC. В настоящее время на рынке доступны стандартные пакеты таких программ, как dBASE II и IV, FoxBase+, Oracle и т.д., позволяющие создавать программы, работающие под контролем сетевой ОС. Фирма NOVELL разработала также ряд уникальных программных продуктов.

## Btrieve

Это система управления базой данных и индексными файлами. Прикладная программа может быть написана на одном из языков высокого уровня (например, Си, Паскаль, Кобол или Бейсик), с которого можно вызывать процедуры пакета Btrieve. С целью оптимизации этот пакет написан на языке Ассемблера. Применение схемы B-дерева в процессе индексации обеспечивает быстрый доступ к данным, т.е. время доступа к информации, записанной на диске, не увеличивается по мере роста базы данных. Разработан сетевой вариант Btrieve и версия для отдельного использования. Данные на диске кодируются. Система предусматривает только парольный доступ к файлам. Кроме того, применены дополнительные средства защиты: файлы доступны лишь для чтения. Только наличие специальных полномочий дает возможность стирать или модифицировать дисковые файлы.

Перечислим некоторые технические данные пакета Btrieve:

- \* число кортежей (records) в файле неограничено;
- \* максимальные размеры файла - 4 Гбайта;
- \* максимальный размер записи - 64 Кбайта;
- \* максимальная длина ключа - 255 байт;
- \* максимальное количество индексов в файле - 24;
- \* количество открытых файлов не ограничивается (15 для Бейсик-интерпретатора);
- \* структура - реляционная база данных. Индексы расположены согласно схеме B-дерева.

## XQL

Это система управления реляционной базой данных. XQL с помощью языка SQL (Structured Query Language) обеспечивает доступ к файлам, находящимся на диске. На практике это сводится к простым вызо-

вам с уровня прикладной программы, которая может быть написана на одном из языков высокого уровня. Пакет обеспечивает большие возможности манипуляции данными на уровне самой базы данных, а также позволяет создавать сложные реляционные структуры, состоящие из нескольких объединенных баз данных или выделенных из них фрагментов. Практическое выполнение упомянутых операций проводится на языке SQL или на уровне так называемых реляционных примитивов.

### Xtrieve

В этом пакете используется сложная система меню, обеспечивающая генерирование рапортов о файлах пакета Btrieve. Он создан для программистов, пишущих прикладные программы (организация "дружественного" интерфейса). Кроме того, пакет Xtrieve является удобным инструментом для пользователей этих программ. Он обеспечивает создание каталогов, анализ, актуализацию и вывод на экран информации, содержащейся в базах данных. Загруженная в систему база может быть представлена на экране в заданном пользователем формате.

Естественным следствием совершенствования компьютеров было создание на локальном уровне систем обмена информацией и доступа к общим базам данных. Следующим этапом была необходимость организации доступа к другим ЛВС и связи между ЭВМ, находящимися на большом расстоянии друг от друга. Программные средства NetWare (NetWare Remote Bridges) дают возможность пользователю включить локальную сеть в структуру ГВС. Здесь есть следующие возможности:

- \* соединение согласно стандарту X.25 (X.25 bridge версия Point-to-Point): соединение 2 ЛВС с помощью модемов, работающих в синхронном режиме, или арендованной телефонной линии. Скорость передачи составляет 64 бит/с. Компьютер, работающий в качестве "моста" (bridge), должен быть оснащен пакетом X.25 Point-to-Point Bridge Software и адаптером PCOX/X.25 (или PCOX/X.25-P);

- \* соединение согласно стандарту X.25 (X.25 bridge версия Multi-Point): соединение до 11 удаленных сетей в одну ГВС. Возможно использование общедоступной (Public Data Network) или другой сети для ретрансляции информации. Каждая из ЛВС должна быть оснащена пакетом X.25 Multi-point Bridge Software и адаптером PCOX/X.25 (или X.25-P), установленными в компьютере, выполняющем роль "моста";

- \* асинхронные соединения (Asynchronous Remote Bridge) – это самое дешевое решение, обеспечивающее соединение 2 или более ЛВС в одну ГВС. Асинхронное соединение может быть выполнено через стандартные порты COM 1/COM 2 или посредством адаптеров WNM (Wide Area Netware Interface Module). Одна карта WNM обеспечивает связь с 4 ЛВС. Программа обслуживания коммуникации между ЛВС (Netware Async. Remote Bridge) может быть установлена в любом из узлов сети. Скорость передачи – 9,6 Кбит/с для стандарта COM 1/COM 2 и 19,2 Кбит/с для адаптера WNM.

Перевод Романа Хмельевского

Более детальную информацию на тему локальных сетей вы можете найти в сборнике "Компьютер-Пресс", в цикле "Локальные сети от А до Я: курс обучения".

С \компьютер на работе\

© Владислав Балинский, Евгений Кривопальцев

# Алиса+

Компьютерный парк в СССР весьма разнороден. Он включает в себя вычислительные средства нескольких направлений и различных поколений (PDP-11, LSI-11, IBM PC/XT/AT, VAX-11, IBM-360(370)), а также аналоги этих компьютеров, произведенные в СССР и в странах СЭВ. Пользователи этого оборудования подчас находятся в весьма затруднительном положении, пытаясь использовать его в различных прикладных задачах, требующих обмена информацией. Кроме того, даже использование "фирменных" сетевых продуктов на базе однородных машин требует наличия "фирменных" сетевых адаптеров. Еще одной стороны проблемы является относительно высокая стоимость современных компьютеров, коммуникационного оборудования и периферийных устройств. Если реализована возможность доступа к оборудованию, принадлежащему какому-либо другому узлу сети, то вовсе необязательно иметь жесткий диск большой емкости или принтер в каждом узле. Кроме сложных проектов, требующих реализации на базе гомогенных или гетерогенных сетей, во многих случаях ставится задача применить уже имеющийся компьютерный парк в качестве периферийного оборудования для более современных машин.

"АЛИСА+" ориентирована на решение указанных проблем. Она разработана коллективом совместного польско-советского предприятия "ПОЛСИБ" и представляет собой программно-аппаратный комплекс для разнородных локальных вычислительных сетей (ЛВС) "АЛИСА+" (удачное название, аббревиатура от "Адаптивная Локальная Иерархическая Сетевая Архитектура". – Примеч. ред.). Предлагаемая вашему вниманию версия ЛВС – принципиально новый этап по отношению к ЛВС "АЛИСА", разработанной в 1984 г. и ориентированной на однородные сети на базе PDP-11, LSI-11. При проектировании новой версии ЛВС были заново разработаны архитектура и сетевые протоколы. Одновременно с развитием программно-математического обеспечения были сделаны серьезные шаги в разработке сетевых адаптеров, в частности, созданы адаптеры типа моноканала – DL-NET и D-NET (в конструкциях QBUS, UNIBUS, IBM PC/XT/AT).

"АЛИСА+" позволяет объединить в сеть компьютеры типа PDP-11 и LSI-11 различных модификаций, работающие под управлением операционных систем RSX11M, RT-11, RSX-11M-PLUS и их аналогов, а также IBM PC/XT/AT, работающие под управлением PC-DOS/MS-DOS. В качестве сетевых адаптеров могут быть использованы среднескоростные надежные интерфейсы D-NET и DL-NET, которые разработал и поставляет "ПОЛСИБ". Одновременно проводятся работы по расширению парка возможных типов ЭВМ за счет включения VAX-11, IBM-360 (370), компьютеров на базе микропроцессоров Intel 80386, Motorola 68030 и использования операционных систем типа UNIX.



Областью применения ЛВС "АЛИСА+" могут быть:

- \* средние уровни промышленных систем (ГПС, АСУП, АСУТП);
- \* системы автоматизации научных исследований;
- \* системы школьной информатики;
- \* различного рода сети ПЭВМ и ВЦКП.

В качестве коммуникационного оборудования допускается применение широкого спектра сетевых адаптеров, в том числе собственной разработки и изготавления. Применяемые адаптеры функционально подразделяются на широковещательные, "точка-точка", мультиплексоры; различаются они по скоростным характеристикам - RS-232, ARCNET, и т.д. Кроме того, "АЛИСА+" допускает простоту адаптации вновь создаваемых сетевых интерфейсов. Таким образом, для создания вычислительных сетей могут быть использованы и простейшие последовательные интерфейсы, и современные сетевые адаптеры моноканалов и транспортных станций (PC NETWORK ADAPTER, D-LINK, ARCNET, ETHERNET, D-NET, DL-NET и т.д.).

Создатели ЛВС "АЛИСА+" при ее разработке придерживались следующих основных принципов.

1. Сетевое обеспечение построено как "незамечаемое" пользователем. Основные идеи - максимальная "дружественность" к пользователю и "растворимость" сетевого продукта в среде применяемой операционной системы. Другими словами, пользователь оперирует только терминами, командами и запросами, стандартными для ОС узла. Системные программы, интегрированные пакеты, базы данных и т.п. могут применять все предоставляемые сетью ресурсы точно так же, как ресурсы собственной ЭВМ.

2. Основным документом, используемым при разработке сетевого обеспечения, является сетевой протокол "АЛИСА+". Принципиально важно, что он не накладывает каких-либо ограничений на применяемые программные средства, что обеспечивает независимость реализаций для различных операционных систем и возможность развития ЛВС на новых объектах.

3. Протокол "АЛИСА+" разрабатывался в соответствии со стандартной 7-уровневой моделью OSI/ISO с учетом некоторых механизмов и возможностей протоколов MAP/TOP, ISO, ECMA. Физический уровень протокола в сетевых адаптерах DL-NET и D-NET выполнен на базе стандарта IEEE 802.4.

4. В зависимости от возможностей (например, объема памяти) и реальных потребностей в конкретном узле (или во всех узлах) могут быть не реализованы некоторые уровни протокола. Так, на конкретном участке сети с моноканальной конфигурацией может отсутствовать сетевой уровень, обеспечивающий связь между несоседними узлами в сетевых конфигурациях типа "точка-точка".

5. Разработанное программное обеспечение для своего функционирования не требует больших объемов оперативной и дисковой памяти. Оно позволяет совмещать работу файлового, сетевого и терминального серверов с работой пользовательских программ. Например, в ОС PC-DOS сетевое программное обеспечение, сконфигурированное на режим максимального сервиса, занимает не более 40 Кбайт оперативной и 300 Кбайт дисковой памяти. При разработке сетевых служб был использован принцип модульности, обеспечивающий их взаимонезависимость.

6. Пользователям ЛВС "АЛИСА+" гарантируется корректность обращения и сохранность информации, доставленной файловому серверу.

7. Пользователям предоставляются средства регистрации и защиты от незаконного доступа.

В состав ЛВС "АЛИСА+" входит ряд служб, обеспечивающих потребности пользователя в обмене информацией и диагностике.

#### \* Служба управления сетью

Предоставляет средства локального и дистанционного управления режимами работы ЛВС, ее параметрами, конфигурацией, распределением ресурсов. Компоненты службы управления позволяют тестировать всю сеть или ее отдельный участок. Отображение информации производится в удобном для пользователя виде и формате.

#### \* Служба доступа к удаленным файлам

Обеспечивает доступ к удаленным файлам из разных узлов сети. Символьные файлы, передаваемые из узлов с различными ОС, приводятся к стандартному виду для ОС узла-приемника. Например, на компьютерах типа IBM PC удаленные файлы располагаются на устройствах, идентифицируемых буквами E,F,G и т.д. Дальнейшая работа с ними как на уровне оператора, так и на уровне любых прикладных и системных программ не отличается от стандартной. Таким образом, можно просмотреть каталог файлов дискового устройства, работающего под управлением ОС RSX-11M, удалить или переименовать на нем файлы, например, с помощью NORTON COMMANDER со стороны IBM PC.

В зависимости от потребностей пользователи могут использовать дисковые или ленточные устройства хранения информации, принадлежащие удаленному узлу сети. Иногда в качестве удаленного файла может выступать печатающее устройство. Данная служба может быть актуальна для "голых" компьютеров. Типичным примером является компьютерный класс, в котором рабочее место обучаемого представляет собой ЭВМ без периферийного оборудования, требующая вызова удаленных файлов, резидентных в рабочем месте преподавателя.

Время доступа к удаленным файлам зависит от типа коммуникационного оборудования, топологии сети и количества конкурирующих потоков информации. Количество одновременно открытых файлов не ограничено и назначается при установке сети.

#### \* Служба вызова удаленных процедур

Выполняет синхронные и асинхронные операции вызова удаленных процедур с передачей параметров. Эта служба может обеспечить задачу, работающую в данном узле, ресурсами, принадлежащими другому узлу сети, например, возможностями арифметического сопроцессора или дополнительными объемами оперативной памяти.

#### \* Служба виртуального терминала

Позволяет присоединить терминал одного узла сети к другому. Это особенно удобно при отладке пользовательского программного продукта, когда узлы сети находятся на значительном расстоянии. Например, узел типа IBM-PC может стать терминалом для ОС RSX-11. В этом случае из него можно произвести необходимые исправления или модификацию и вновь запустить отлаживаемый комплекс программ. При использовании IBM PC в качестве виртуального терминала для ОС RSX-11M или RT-11 можно выбрать режим эмулируемого терминала: VT-52 либо VT-100.

Служба виртуального терминала является также удобным средством перераспределения терминальных средств между узлами сети без их фактической перекоммутации.

#### \* Служба электронной почты

Обеспечивает передачу сообщений пользователей между узлами сети.

В заключение следует сказать, что ЛВС "АЛИСА+" прошла опытную эксплуатацию на ряде советских предприятий на оборудовании различной конфигурации.



# Программы-упаковщики

В последние годы на Западе настоящий ажиотаж вызвало появление так называемых программ упаковки (уплотнения) данных. Их популярность связана также с широким использованием телефонных линий для трансмиссии данных. Благодаря уплотнению можно сократить время (а значит, и уменьшить стоимость) трансмиссии на 10-80%. Подобное процентное расхождение является следствием того, что не все файлы поддаются упаковке с одинаковой эффективностью. Относительно хорошо сжимаются текстовые файлы, в среднем их объем можно уменьшить на 40-60%, в случае же скомпилированных программ степень уплотнения составляет 20-40%. Некоторые файлы с небольшой избыточностью почти совсем "некомпрессируются" (примером могут служить архивные файлы, создаваемые упаковочными программами).

Программы-упаковщики, как правило, располагают возможностями значительно выходящими за рамки обычного уплотнения файлов. Так, все имеющиеся на рынке современные упаковщики дают возможность создания "архивов", содержащих большое число упакованных файлов; в результате получается двойной эффект. Во-первых, на диске освобождается отведенное для них место (что является весьма существенным, особенно в том случае, когда мы имеем дело с большим количеством коротких файлов). Во-вторых, упаковка некоторой программы с различными вспомогательными файлами в одном архиве значительно облегчает такие операции, как перенос на другую систему, создание копий и т.д. Например, на фирменных дисках популярного компилятора Паскаль фирмы BORLAND (TurboPascal) программы и данные находятся в сжатом виде (файлы с расширением .ARC). В версии 5.5 этой программы упакованы не только демонстрационные программы, но и драйверы, документация стандартных библиотек и вспомогательные программы.

Большинство существующих программ позволяет записывать относительные дорожки доступа к сжатым файлам, присоединять краткие описания как к отдельным файлам, так и ко всему архиву, легко актуализировать содержимое архива и т.д. Все известные мне программы реализуют следующие основные функции:

**ADD** - добавление файла(ов) в архив. Существует возможность использовать в именах шаблоны (\* и ?).

**MOVE** - присоединение файла(ов) к архиву и последующее стирание их с диска. Стирание происходит после завершения упаковки и проверки созданного архивного файла.

**UPDATE** - присоединение файла(ов) к архиву только в том случае, когда в архиве нет файла с таким же именем, либо он есть, но старше, чем файл на диске.

**FRESHEN** - присоединение файла(ов) к архиву только в том случае, когда файл с таким же именем со-

держится в архиве, но он старше, чем файл на диске.

**LIST (VIEW)** - обзор содержимого архива.

**DISPLAY** - распаковка файла(ов) и пересылка на стандартное выходное устройство.

**PRINT** - распаковка файла(ов) и пересылка на печатающее устройство или на последовательный интерфейс.

**EXTRACT** - распаковка файла(ов) на диск.

**DELETE** - удаление файла из архива.

**SELF-EXTRACT (SFX)** - образование из содержимого архива исполняемого файла (.EXE), который, после запуска, распаковывает сам себя. Это - популярный метод распространения программ shareware и freeware вместе с документацией, так как для распаковки таким образом созданного архива не нужны дополнительные программы. Обычно этот способ упаковки обеспечивается дополнительной программой, прилагаемой к основному пакету программ.

Приведенные выше ключевые слова-функции, как правило, отражаются в документации к программам и указаны здесь только для того, чтобы облегчить пользование ими.

У данных функций почти всегда есть модификаторы и дополнительные параметры, облегчающие пользование программой, помогающие создавать описания, форматировать распечатки и т.д.

Используются преимущественно три формата уплотнения: **ARC/PAK**, **LZH** и **ZIP**. Существуют, правда, и другие форматы (например, **ZOO**, **DWC** или **LZS**), но они менее популярны. Свои названия форматы получили от расширений, присваиваемых архивным файлам отдельными программами.

Формат **ARC** был разработан фирмой **SYSTEM ENHANCEMENT ASSOCIATES, Inc. (SEA)**, которая в 1985 г. выпустила программу **ARC**, обслуживающую файлы такого формата.

В июне 1986 г. фирмой **PKWARE** был выпущен пакет быстродействующих программ **PKArc/PK(X)Arc**, также использующих этот формат. Пакет был полностью совместим с программой **ARC**, но мог использовать новый метод уплотнения. Оригинальный **ARC** определял новый формат файла, но был не в состоянии его распаковывать. Из-за проигранного фирмой **PKWARE** процесса о нарушении авторских прав, касающихся формата **ARC** и торговой марки **ARC**, последняя версия **PK(X)Arc** (3.61, 1989 г.) появилась под названием **PKPak/PKunPak** и больше уже не модифицировалась.

Следующей модификацией формата **ARC** является формат **PAK** фирмы **NOGATE** и отличающийся от **ARC** только новым методом уплотнения. Этот формат обслуживается программой **PAK** (фирмы **NOGATE**), использующей все старые методы упаковки формата **ARC** до **PKPAK** включительно.

Формат **LZH** и программа **LHARC** являются "детищами" японского автора Аруясу Есизаки (Haguyasu Yoshizaki) и были разработаны им в 1988 г. Некоторое время они успешно конкурировали с **ARC** из-за более высоких возможностей упаковки. Как сам формат, так и программа общедоступны (freeware).

Формат **ZIP** (1989 г.) - триумфальное возвращение на рынок **PKWARE**. После проигранного судебного процесса фирма опубликовала новый формат упаковки, снабдив его пакетом отличных обслуживающих программ. Сам формат раскрыт и также общедоступен (freeware).

Программы-упаковщики тестировались путем упаковки трех групп файлов: индексных и вспомогательных файлов **dBBase** (файлы **.DBT**, **.FMT**, **.NDX** и **.PRG**



общим объемом 176562 байта), выполняемых программ .EXE (использовалась часть программ, входящих в состав пакета Norton Integrator общим объемом 500039 байт) и текстовых файлов .TXT с общим объемом 500049 байт. Отмечалось время упаковки и распаковки, размер получаемого файла, размер файла

SFX и время распаковки архива SFX. Однако результаты тестирования я представлю в несколько ином виде: скорость упаковки (компрессии, COM) и распаковки (декомпрессии, DEK) - в Кбайтах неуплотненного файла на секунду; сэкономленное место (PAK) - в процентах (0% - отсутствие уплотнения); величина модуля SFX (константа для данной программы, не зависит от архива) - в байтах; скорость самоупаковки ар-



**ARC.EXE** - очередная версия ARC под номером 6.02 (январь 1989 г., 65 339 байт). В этой версии устранены некоторые ошибки, повышенено быстродействие, добавлен новый метод сжатия (используемый PKARC и PAK). Кроме того, есть возможность запоминания дорожек доступа к файлам. Программа ARC реализует все вышеупомянутые функции, кроме образования SFX (самораспаковывающихся программ). Однако в дистрибутивном файле (ARC\_602.EXE, 129741 байт) содержится программа MKSARC.EXE (18831 байт), реализующая эту задачу. Особенностью ARC является возможность запуска программы непосредственно из архива (без распаковки на диск).

**ARCA.COM** и **ARCE.COM** - это две программы, авторами которых являются Уэйн Чин (Wayn Chin) и Вернон Д. Берг (Vernon D. Buerg). Первые их версии появились в начале 1986 г. В настоящее время используются ARCA v.1.29 (декабрь 1987 г., 4412 байт) и ARCE v.4.0 (март 1989 г., 7517 байт). Это специализированные программы, реализующие только основные функции: ARCA присоединяет файлы к архиву, используя лишь два метода из девяти, а ARCE распаковывает архив, распознавая все примененные методы сжатия. Применяются они довольно часто и в основном из-за своих небольших объемов. Величина дистрибутивных файлов зависит от метода сжатия и составляет около 10 Кбайт. В дистрибутивном архиве ARCA дополнительно имеется версия упаковки для процессора 80286.

**PKARK.COM** и **PKXARC.COM**. Автор этих программ - Фил Кац (Phil Katz) из фирмы PKWARE. Первые версии появились в июне 1986 г. Последняя - 3.61 появилась под названием PKPAK.EXE (21720 байт) и PKUNPAK.EXE (15112 байт). PKPAK является инструментом для образования архива и исправления в нем ошибок, а PKUNPAK применяется для распаковки файлов. Данный пакет обладает весьма широкими возможностями. Интересным (но малозелегантным) является метод образования программ SFX: из дистрибутивного файла специальная программа MAKESFX.COM создает новый файл под названием PKSFX.PRG. Самораспаковывающаяся программа SFX образуется с помощью простого под克莱ивания этого файла в начало архива и присвоения имени с расширением .EXE. В настоящее время эти программы не развиваются.

**PAK.EXE** была создана фирмой NOGATE CONSULTING в октябре 1988 г. Программа использует формат, соответствующий ARC, но архивные файлы имеют другое расширение (.PAK). В программу добавлен новый (уже десятый) метод сжатия. Самой последней версии програм-

мы (2.10) у меня еще нет, поэтому я протестирую более раннюю версию 1.6. Однако это тестирование может оказаться не совсем точным, так как традиционно изменение первой цифры номера версии обозначает, как правило, серьезные изменения в работе программы. Так же, как и ARC, пакет PAK v.1.6 содержит две программы: PAK.EXE (60176 байт) и EXEMAKE.EXE (21456 байт). Первая из них выполняет почти все действия, связанные с архивными файлами, вторая - служит инструментом для образования архивов SFX. Как отмечает сама фирма NOGATE, она не ставила целью достижение быстродействия PK(X)ARC, а стремилась максимально увеличить степень уплотнения. Проведенный тест полностью это подтвердил. Новым в программе PAK является то, что пользователь может сам определить значение передаваемых программе параметров и комментариев.

**LHARC.EXE** относится к совершенно другому семейству. Она была написана Аруясу Есизаки в 1988 г. Он использовал собственный формат архива и новый метод сжатия. Результатом его работы явилась повышенная эффективность за счет относительно невысокого быстродействия. Последняя версия - 1.13c (май 1989 г., LH13C.EXE, 36963 байта) - объединяет в одной программе (LHARC.EXE, 30470 байт) все функции, включая образование трех видов архивов SFX. Здесь можно отметить несколько особенностей: автоматический выбор типа архива SFX (COM или EXE), возможность образования и распаковки файла SFX, значительно превышающего объем доступной памяти, а также автоматический запуск содержащихся в SFX программ. В последней версии, по примеру PKZIP, добавлена возможность подхода к SFX как к обычному архиву.

**PKZIP.EXE** и **PKUNZIP.EXE** - пакет упаковочных программ фирмы PKWARE. Первая, "допремьерная" версия 0.90, появилась на рынке в 1989 г.; последняя действующая версия уже имеет номер 1.02. PK(UN)ZIP предлагает два метода уплотнения: один - оптимизирующий степень сжатия, второй - быстродействие. PK(UN)ZIP сам определяет наличие процессора 80386 и использует его возможности. Программа "умеет" запоминать дорожки доступа к упакованным файлам, делать описание всего архива, а также отдельных файлов и т.д. Архивы SFX образуются с помощью прилагаемой к пакету программы ZIP2EXE.EXE (6906 байт). К нему также добавлена программа, позволяющая устранять некоторые повреждения архивных файлов (PKZIPFIX.EXE, 8926 байт). PK(UN)ZIP в состоянии правильно обрабатывать архивы SFX (большинство программ не опознает их как архивы).

## 1. ARC v.6.02, длина модуля SFX - 7726 байт

	dBase	EXE	TXT
KOM	3.19	183	2.06 [KB/s]
DEK	4.21	274	2.40 [KB/s]
PAK	70.57	25.6	42.79 [%]
SFX	5.22(*)	3.37	2.92 [KB/s]

## 2. ARCA v.1.29, ARCE v.4.0c, отсутствие модуля SFX

	dBase	EXE	TXT
KOM	3.67	2.08	2.20 [KB/s]
DEK	5.07	3.39	2.82 [KB/s]
PAK	70.94	26.12	43.83 [%]
SFX	-	-	- [KB/s]

## 3. PK(UN)PAK v.3.61, модуль SFX: 9758 байт (параметр -o: совместимость с ARC)

	dBase	EXE	TXT
KOM	6.38	3.83	4.63 [KB/s]
DEK	6.16	4.65	3.54(*) [KB/s]
PAK	70.92	26.69	43.86 [%]
SFX	5.22(*)	3.91	3.32(-) [KB/s]

## 4. PK(UN)PAK v.3.61, модуль SFX: 9758 байт

	dBase	EXE	TXT
KOM	6.90(*)	4.28(*)	4.83 [KB/s]
DEK	5.95	4.78	3.36 [KB/s]
PAK	71.63	27.12	44.23 [%]
SFX	5.07	4.28	3.36(*) [KB/s]

## 5. PAK v.1.6, модуль SFX: 6742 байта (параметр /S: Squashing)

	dBase	EXE	TXT
KOM	2.39	1.95	2.05 [KB/s]
DEK	2.50	2.07	1.84 [KB/s]
PAK	71.41	26.67	43.92 [%]
SFX	2.97	2.31	1.99 [KB/s]

хива SFX (SFX) - в Кбайтах неуплотненного файла в секунду.

Подведем итоги. Наиболее слабой оказалась PAK. Правда, ее результаты лучше, чем у ARC, но она далеко позади PKZIP и LHARC. Из игры также выбывают программы ARC и ARCA/ARCE. С точки зрения быстродействия на первое место выходит PKPAK. Она выполняет упаковку очень быстро и почти независимо от вида файла. Однако PKZIP уступает ей в быстродействии очень немного, в то время, как размер получаемого файла оказывается намного меньше (степень уплотнения намного выше, иногда даже в два раза).

Что касается скорости распаковки, то PKUNZIP не имеет себе равных. С точки зрения эффективности упаковки первое место делят между собой LHARC и PKZIP (с очень небольшим разрывом в результатах). Однако PKZIP упаковывает файлы в два раза быстрее, чем LHARC.

Скорость распаковки архивов SFX мало дифференцирована и трудно определить, какая из программ лучше. По остальным результатам измерений на первое место выходит PK(UN)ZIP. При упаковке, оптимизированной по скорости, она немного уступает другим программам, образуя архивы почти такой же величины. По степени компрессии она уступает (но немного и не всегда) только LHARC, однако оставляет ее далеко позади в отношении быстродействия. PKUNZIP быстрее всех распаковывает свои архивы, незначительно опережая PKPAK.

С учетом всего сказанного, а также того, что развитие программ PK(X) ARC/PK(UN)PAK уже завершено, становится понятно, почему PKZIP пользуется в мире такой популярностью. При тестировании на оборудовании более высокого класса (я проводил тест на XT) перевес PK(UN)ZIP был бы еще больше, так как она распознает (и в состоянии использовать) процессоры 80286/386.

## 6. PAK v.1.6, модуль SFX: 6742 байта (параметр /C: Crushing)

	dBase	EXE	TXT
KOM	2.24	1.30	1.67 [KB/s]
DEK	2.43	1.54	1.55 [KB/s]
PAK	74.63	32.85	44.25 [%]
SFX	2.78	1.66	1.69 [KB/s]

## 7. LHARC v.1.13c, модуль SFX: 1295 байт

	dBase	EXE	TXT
KOM	0.35	0.93	0.92 [KB/s]
DEK	2.65	1.46	1.46 [KB/s]
PAK	80.29	42.20(*)	51.21(*) [%]
SFX	3.67	1.87	1.98 [KB/s]

## 8. PK(UN)ZIP v.1.02, модуль SFX: 15512 байт (параметр -ex: Inflating)

	dBase	EXE	TXT
KOM	1.30	1.80	1.86 [KB/s]
DEK	6.39(*)	5.74(*)	3.5(-) [KB/s]
PAK	81.63(*)	42.06	50.16 [%]
SFX	4.54	4.43(*)	2.97 [KB/s]

## 9. PK(UN)ZIP v.1.02, модуль SFX: 15512 байт, (параметр -es: Shrinking)

	dBase	EXE	TXT
KOM	5.95	3.970	5.03(*) [KB/s]
DEK	5.22	4.138	3.11 [KB/s]
PAK	72.89	27.79	43.41 [%]
SFX	4.11	3.439	2.74 [KB/s]

Звездочкой (\*) обозначен лучший результат в данной категории, а минусом (-) - результаты, расположенные в пределах плюс/минус 10%.

Теперь о достоверности представленных результатов. Я должен признаться, что были случаи, когда LHARC упаковывала архивы гораздо лучше, чем PKZIP, но иногда все получалось наоборот. Для меня убедительным доказательством эффективности LHARC или PKZIP послужил, например, тот факт, что они могут (хотя и не всегда) уплотнять архивы PKPAK на 5-10% (обратных случаев я не обнаружил).

Все рассмотренные программы работают только с IBM PC или с машинами, использующими систему MS-DOS. Однако проблема уплотнения данных касается не только IBM. Мне известны случаи, когда ARC, LHARC и PKUNZIP работали на ATARI ST (в последнем случае обеспечивая возможность переноса данных из MS-DOS; программы PKZIP на ST я не встречал).

Исследованные программы распространяются как shareware и freeware (программа LHARC только как freeware). Эти формы отличаются друг от друга тем, что за программное обеспечение shareware пользователь должен платить только после того, как опробует программу и придет к выводу, что она полностью отвечает его требованиям. В обоих случаях программы можно копировать и распространять при условии, что это делается бесплатно и что распространяется исключительно оригинальный дистрибутивный пакет. В любом случае нельзя продавать или распространять модифицированную версию программы, неполную или поврежденную программу и т.д. Нельзя также включать в эти программы собственные дополнения без письменного согласия автора.

К каждой программе вместе с регистрационным формуляром прилагаются гарантии, а также разрешение на пользование и распространение. Ни в коем случае нельзя распространять программы без этих документов.

Перевод Тадеуша Радюша

\Компьютер на работе\

© Мартиш Дец, Марек Матушак, Марек Млынарский

# Компьютерный склад информации

## Гибкий диск - физическое описание...

Сегодня невозможно представить себе работу на компьютере без этой небольшой круглой пластины из специального гибкого пластика, покрытого магнитным слоем. А ведь история этого носителя информации началась сравнительно недавно. Первые диски появились в начале 70-х годов и в связи со своими многочисленными достоинствами начали быстро вытеснять другие методы регистрации компьютерной информации.

До эпохи гибких дисков безраздельно царили магнитные ленты, имевшие один крупный недостаток, - большое время доступа к информации. Проблема быстрого доступа к нужным данным хорошо знакома каждому, кто имел дело с двумя устройствами: магнитофоном и его противоположностью в рассматриваемом аспекте - граммофоном. Именно идеей структуры грампластинок воспользовались создатели дисководов и самих гибких дисков (floppy disk).

В накопителе дискета вращается со скоростью 300-360 оборотов в минуту, а головка чтения/записи информации перемещается в радиальном направлении. В совокупности это позволяет почти немедленно достичь любой точки на рабочей поверхности диска.

Гибкие диски выпускаются главным образом двух размеров: диаметром 3,25 дюйма (около 13 см) и 3,5 дюйма (около 9 см). В компьютерах старых типов встречаются 8-дюймовые дискеты (около 20 см). В некоторых компьютерах (в том числе Amstrad и Spectrum) применяются дискеты диаметром 3 дюйма, на выставках и ярмарках можно встретить дискеты диаметром 2 и 2,5 дюйма. В зависимости от качества магнитного слоя (основой которого является окись железа) данные могут записываться с одной или с обеих сторон дискеты. Это зависит также от типа НГМД, установленного в компьютере. От конструкции накопителя зависит также количество записываемой на дискету информации (двусторонний гибкий диск 3,5 дюйма может содержать до 1,44 Мбайт информации, т.е. около 720 машинописных страниц).

Пластиковый диск покрыт магнитным слоем толщиной 2,2-2,9 микрометра. Он помещен между двумя слоями мягкой прокладки, по которой скользит во время вращения, и вместе с ними вложен в полужесткий (или жесткий в случае дискет диаметром 3 и 3,5 дюйма) конверт.

Конверт 3,25-дюймовой дискеты имеет следующие отверстия: для шпинделя привода, для доступа головки чтения/записи к диску, индексное отверстие (позволяющее дисководу точно определить расположение секторов) и вырез для маркера защиты от записи. Отверстие для привода в самой дискете иногда усилено специальным кольцом (hardhole), но практика показывает, что это не имеет решающего значения.

Если вырез маркера защиты закрыт, то накопитель не может осуществлять записи; состояние выреза автоматически проверяется перед ее началом. В 3,5-дюймовых дискетах применяется жесткий конверт, а отверстие для головки закрыто специальным щитком. Отверстие для привода находится только с одной стороны, а маркер защиты от записи встроен в конверт и действует "наоборот" - открытое отверстие маркера блокирует запись.

Продолжительность жизни гибких дисков в значительной степени зависит от нашей осторожности и правильного использования. Главные враги дисков - это грязь и пыль, а также табачный дым. Записанную на диске информацию можно утратить, даже не заметив этого, если диск будет подвергнут воздействию магнитного поля (трансформаторы, телефоны, телевизоры, электродвигатели). Следует соблюдать осторожность, проходя через металлоискатели в аэропортах. Сотрудники нашей редакции в таких случаях стараются передавать коробки с дискетами на ручной контроль.

Правильно эксплуатируемый диск выдерживает несколько миллионов проходов головки по дорожке, что означает несколько месяцев непрерывной работы на одной дорожке, а ведь таких дорожек на диске несколько десятков. Дискеты высокого качества известных и опытных изготовителей гарантируют в среднем 70 млн. проходов головки по дорожке, что на практике сводится к более чем 20-летней интенсивной эксплуатации. Теоретически можно скорее ожидать аварии головки или других механизмов накопителя. Практика же показывает, что иногда дискеты имеют почаще всего безымянное (name), т.е. сми неизвестной реальности. Для информации не слепить только на вредления, и это случается с ми изделиями (помощью дешевыми дискавестных фирм. В практике мы уже что на носителях экономит и надо покупать высококачественные изделия.

Во время записи информации головка намагничивает магнитные частицы магнитного слоя, обретают одинарность. Для записи бита намагниченного отрезок длиной 0,004 мкм и шириной 0,015 мкм. Магнитный слой стабильность вечно, хотя практически можно вить себе изменившиеся дестабилизации магнитной линии Земли. Во всяком случае магнитные ленты с записанными в середине 30-х годов можно и сегодня воспроизводить без потери качества, а ведь они являются примером первых попыток магнитной записи информации.

Для поддержания чистоты головок накопителей и продления жизни дисков применяются специальные чистящие диски. Они покрыты специальным материалом, который снимает с головок эксплуатационные загрязнения. Очистку головок стоит производить довольно часто - до 3-4 раз в месяц. Лучше всего просто купить специальную дискету и регулярно ею пользоваться.

С точки зрения взаимодействия ЭВМ с носителем информации, т.е. с диском, чрезвычайно важен метод записи и чтения информации. Он лишь частично связан с описанными выше физическими свойствами дискеты, а в основном зависит от конструкции адаптера и НГМД, а также от операционной системы.

### ...и как он записывается

Способ записи данных на дискете зависит от вида накопителя, от его контроллера, а также от операционной системы, под контролем которой работает ЭВМ. Что же это такое - контроллер НГМД?

Еще несколько лет назад можно было бы спокойно написать, что систем записи на магнитном носителе почти столько, сколько фирм, выпускающих компьютеры, и напомнить, что когда создатели микрокомпьютера Apple Возняк и Джобс решили применить в нем дисковод, то они в течение одной ночи разработали дисковый контроллер. Кстати, это была чрезвычайно интересная конструкция со скоростью вращения, изменяющейся в зависимости от номера дорожки (постоянная линейная скорость записи). К счастью, эти бурные времена уже позади, и можно ограничиться двумя самыми распространенными способами записи: **FM** (frequency modulated) и **MFM** (modified frequency modulated).

В записи FM соседние биты данных всегда отделены импульсом синхронизации. Благодаря этому снижаются требования к качеству магнитного носителя. К сожалению, снижается и объем хранимой на нем информации.

Удвоение этого объема возможно при записи MFM, где импульс синхронизации записывается на диск только тогда, когда два бита данных подряд имеют нулевое значение. Из известных машин такая форма записи используется в компьютерах IBM PC, Amstrad, MSX, Atari ST, возможна она в НГМД VC-1570/1571 Commodore и в дисковых операционных системах TIMEX. Часто встречающиеся определения "одинарная плотность записи" и "двойная плотность записи" означают нечто иное, как именно FM и MFM. Из уважения к фирме Commodore упомянем еще об одном виде записи - **GCR** (Group Code Recording). Он позволяет еще более увеличить плотность записи и применяется в дисководах для домашних компьютеров Commodore.

### Что происходит во время форматирования дискеты?

Первая операция, выполняемая на новом диске - это форматирование. Этот процесс позволяет придать диску его окончательную физическую структуру. В ходе форматирования определяется, в частности, количество дорожек, число секторов на дорожке и длина секторов. Количество дорожек обычно соответствует номинальному числу дорожек дисковода, а типичная зависимость между числом секторов на дорожке и длиной сектора (т.е. числом байт в секторе) представлена в табл. 1.

Длина считываемого или записываемого сектора обычно указывается путем загрузки соответствующего значения в один из регистров микропроцессора контроллера НГМД.

Таблица 1

Типичная зависимость между числом секторов на дорожке и длиной сектора

Число секторов на дорожке	Длина сектора	
	ФМ	МФМ
5	512	1024
9	256	512
16	128	256

### Логическая организация, даже для дискеты?

В операционной системе MS-DOS предусмотрены четыре логических области дискеты:

- загрузочный сектор;
- таблица размещения файлов - ТРФ;
- каталог;
- область данных.

Загрузочный сектор содержит короткую программу начальной загрузки ОС в память машины. Независимо от формата записи, эта программа всегда занимает один сектор - нулевой сектор нулевой дорожки на нулевой стороне диска. В следующих секторах расположена таблица размещения файлов (ТРФ). Она содержит информацию, где в области данных находятся записанные на дискету файлы. Отметим, что соседние фрагменты файла отнюдь не обязательно записываются в соседних секторах. Новые файлы могут занимать место, освобожденное в результате стирания ранее записанных. В связи с большим значением информации, хранящейся в ТРФ, на дискете находятся две одинаковые таблицы. Непосредственно за таблицей размещения файлов находится каталог. В нем записаны основные параметры (например длина) файлов, записанных в области данных.

Величина области данных, каталога и ТРФ зависит от числа секторов на дискете, которое в свою очередь обусловлено форматом записи данных. В ОС MS-DOS длина сектора составляет 512 байт, но число секторов может быть различным и зависит от версии системы и типа накопителя (см. табл. 2).

Таблица 2

Сравнение основных форматов ОС MS-DOS

Формат	Число секторов на диске	Размер ТРФ	Размер каталога (две копии)	Размер области данных
S-8	320	2	4	313
D-8	640	2	7	630
S-9	360	4	4	351
D-9	720	4	7	708
QD-9	1410	10	7	1428
QD-15	2400	14	14	2371

В столбце "Формат" обозначено количество дорожек и сторон диска (S - одна сторона, 40 дорожек; D - две стороны, 40 дорожек; QD - две стороны, 80 дорожек). Цифры определяют число секторов на дорожке. Размеры логических областей дискеты даны в секторах.

Продолжение читайте в следующем выпуске.

Перевод Андрея Поплавского

# Выбор принтеров

В предыдущей статье мы вкратце описали полный процесс подготовки издания на персональном компьютере, а также перечислили необходимые для работы технические и программные средства. Эта статья посвящена выбору принтеров для настольной издательской системы. При правильно поставленной технологии подготовки издания необходимо несколько принтеров. Даже в настольных издательских системах, имеющих в своем комплекте фо-

Мы не будем останавливаться на "ромашковых" принтерах из-за низкой скорости печати (от 10 до 60 символов/с) и отсутствия графических возможностей, а главное - неприемлемой системы смены шрифтов. Из-за всех этих недостатков "ромашковые" принтеры вряд ли получат большое распространение в издательских системах.

Наиболее широко применяются матричные принтеры. В матричных принтерах печатающая головка представляет собой матрицу, состоящую из фиксированного числа иголочек (ріп). Качество печати принтера определяется количеством иголочек по вертикали головки, и в соответствии с этим матричные принтеры можно разделить на 9-, 15-, 18- и 24-игольные (некоторые принтеры допускают установку сменных головок). Лучшее качество в принтерах с большим числом иголочек достигается за счет более плотного их расположения в печатающей головке (таким образом на единицу площади приходится большее число печатных точек). При выборе матричного принтера кроме качества печати и стоимости необходимо учитывать и ряд других характеристик. Весьма важная характеристика (иногда определяющая выбор) - наличие в принтере внутренней памяти, достаточной для загрузки национальных шрифтов (вопрос разработки шрифтов для принтеров различных типов будет рассмотрен в наших будущих статьях). Сейчас скажем только, что загружаемые шрифты представляют собой файлы специального формата на внешних устройствах, хранящие образы букв, которые могут быть переданы в специальный буфер принтера для дальнейшего использования при печати. В принтере уже есть несколько наборов шрифтов и микросхема может быть перепрограмми-

тонаборное оборудование, без принтера не обойтись. Выбор принтеров - часть задачи комплектации оборудования и программных средств, поскольку важно обеспечить совместимость программных средств и оборудования. Решающим требованием к принтерам является возможность работы с ними выбранной вами программы верстки и текстового процессора. Программы верстки и текстовые процессоры поддерживают работу с большим набором принтеров.

По принципу действия принтеры делятся на контактные и бесконтактные. В первых изображение получается за счет соприкосновения печатающей головки с бумагой через красящую ленту. К принтерам этого типа относятся, в частности, матричные и "ромашковые" принтеры. В бесконтактных принтерах нет непосредственного соприкосновения бумаги с печатающей головкой или печатающей головкой отсутствует вообще. Бесконтактные принтеры можно разделить на струйные и лазерные.

Хорошая конфигурация аппаратных средств настольной издательской системы должна включать принтеры различных типов, так как на разных этапах подготовки издания можно воспользоваться печатной копией различного качества. Репродуцируемый оригинал-макет получают на лазерных принтерах, которые существенно дороже матричных (дороже обходятся и комплектующие материалы для них). Для сверки же текста, подготовленного в текстовом процессоре, вполне подойдет качество печати матричного или струйного принтера. Выходом настольной издательской системы, как мы уже говорили в предыдущей статье, не обязательно может быть оригинал-макет. Во многих случаях вам нужно будет создавать документы с более "мягкими" полиграфическими требованиями. Для этой цели вполне подойдет матричный принтер. При выборе принтера кроме стоимостных соображений первостепенное значение имеет возможность использования в них кириллицы.

рода - на на русский алфавит. Выбранный вами принтер должен допускать тот или иной вариант работы с национальным шрифтом. Некоторые фирмы поставляют специальные кассеты, содержащие дополнительные шрифты и нужный вам набор шрифтов можно заказать фирм-поставщику.

Следующая характеристика, которую надо учитывать - скорость печати принтера, измеряемая в количестве печатных символов в секунду. Скорость печати матричного принтера зависит от режима печати. Существуют три режима - Draft, NLQ (Near Letter Quality) и LQ (Letter Quality). В зависимости от числа иголочек принтера он может допускать один или два из этих режимов. Режим печати определяет качество изображения символа. В режиме Draft символ печатается за один проход печатающей головки, поэтому размер матрицы символа будет соответствовать размеру матрицы головки. В режиме NLQ символ печатается за два прохода головки с ее вертикальным смещением, и обычно размер матрицы увеличивается вдвое, что позволяет улучшить качество печати, но одновременно с этим снижается скорость. Режим LQ аналогичен режиму NLQ, однако обеспечивает более высокое качество печати. Печатающие устройства с небольшим числом иголочек допускают, как правило, режимы Draft и NLQ, а принтеры с большим числом - Draft и LQ. Принтеры различаются и графическими возможностями. Более дешевые дают меньшее разрешение в графическом режиме.

Современные печатающие устройства имеют довольно сложную систему управления, позволяющую изменять выбранный шрифт (увеличивать или уменьшать размер или толщину буквы), печатать текст с разной плотностью как по вертикали, так и по

горизонтали, а также со сдвигом на полстроки вверх или вниз, подчеркивать текст. Принтеры с большим разрешением, кроме того, могут иметь несколько встроенных шрифтов различных начертаний; практически все современные матричные печатающие устройства включают курсив.

При выборе матричного принтера необходимо учитывать размер печатных документов, для вывода которых он предназначен. Принтеры различаются шириной каретки (узкая, широкая). В большинстве случаев фирмы выпускают модификации одного и того же принтера на разную ширину каретки. Следующая характеристика принтера, которую необходимо учитывать при выборе, — возможность автоматической подачи бумаги. Устройство автоматической подачи листовой бумаги (sheet feeder) иногда входит в комплект принтера, но в основном оно поставляется за отдельную плату и должно быть отдельно указано в заказе. Устройства автоматической подачи бумаги различаются емкостью (в листах) подающего механизма и форматом бумаги. При выборе принтера следует четко спланировать, какое место он будет занимать в технологическом процессе и к какому персональному компьютеру будет подключен. Принтеры могут иметь плату параллельного и/или последовательного интерфейса. Соединительный кабель для принтера зачастую должен быть указан отдельно в спецификации на поставку. Многие принтеры могут эмулировать работу нескольких принтеров, т.е. воспринимать команды различных принтеров. Допустимые режимы работы принтера надо рассматривать с учетом выбираемого вами программного обеспечения. Еще одна характеристика, которую хотелось бы упомянуть, — объем буфера принтера, используемый для промежуточного хранения данных перед выводом на бумагу. Чем больше буфер принтера, тем меньше времени процессор тратит на внутренние переключения. Эта характеристика влияет на общую производительность системы. Последняя характеристика, которая может быть принята во внимание, — возможность цветной пе-

чати.  
Существует  
много разновиднос-  
тей цветных матричных принте-  
ров.

Альтернативой матричным принтерам могут быть бесконтактные струйные принтеры. В струйных принтерах для получения изображения используются различные электрические способы нанесения чернил на бумагу. Стоимость и скорость струйных принтеров сравнима со стоимостью и скоростью матричных 24-игольных принтеров. К достоинствам струйных принтеров можно отнести бесшумность их работы, компактность, наличие большего, чем в матричных, числа встроенных шрифтов, возможность расширения внутренней памяти для загрузки пользовательских шрифтов и довольно высокое качество печати. Однако при покупке струйного принтера необходимо учитывать, что они более чувствительны к качеству бумаги.

Лазерные принтеры обеспечивают наивысшее качество печати. В них используется специальная методика напыления и закрепления точек изображения, разработанная фирмой CANON. Эта методика позволяет получить высококачественную печать, которую можно применять в качестве репродуцируемого оригинал-макета. Цена принтера прямо зависит от его разрешения (число точек на дюйм). Наиболее простые лазерные принтеры имеют разрешение 300 точек/дюйм, более сложные — 600 точек и выше. При выборе лазерного принтера прежде всего следует учи-

тывать полиграфические требования к оригинал-макету. Такие принтеры, как правило, позволяют эмулировать работу двух-трех принтеров, имеют платы как последовательного, так и параллельного интерфейса. Скорость печати измеряется для лазерных принтеров в листах/мин. Средние принтеры печатают со скоростью 6-8 листов/мин.

Очень существенная характеристика лазерного принтера, которую надо учитывать при его приобретении, — объем внутренней памяти. Как правило, в стандартной поставке лазерный принтер содержит только 512 Кбайт памяти. Для настольной издательской системы этого явно недостаточно, так как при таком объеме памяти резко сокращаются возможностях вывода графического материала — рисунков, фотографий и т.д. Поэтому при заказе лазерного принтера необходимо отдельно указать расширение памяти до 2-4 Мбайт. Лазерные принтеры имеют значительно большее количество встроенных шрифтов высокого качества и позволяют загружать ряд шрифтов различных гарнитур одновременно (30 и более шрифтов). Переключение между шрифтами осуществляется командами принтера из программ верстки или текстовых процессоров. Разработка шрифтов для лазерных принтеров — отдельная и очень трудоемкая задача, требующая соответствующей технологии (матрица буквы состоит из нескольких десятков точек). Перепрограммировать встроенные шрифты в обычных условиях невозможно. Встроенные русские гарнитуры можно заказать только фирме-поставщику, если она выполняет этот род работ, поэтому с лазерными принтерами используются в основном загружаемые русские шрифты. Для этих принтеров также поставляются отдельные кассеты с дополнительными шрифтами. Лазерные принтеры принимают только листовую бумагу различных форматов, печать на которой может осуществляться по длине (режим Portrait) или ширине (режим Landscape) листа. Бумага должна быть высокого качества с большим содержанием древесины. Лазерные принтеры также бывают черно-белыми и цветными. При выборе лазерного принтера следует учитывать возможность работы с языком

PostScript. Для ряда принтеров плата работы с языком PostScript должна быть отдельно оговорена в спецификации. Лазерный принтер — обязательная

составляющая настольной издательской системы. Для того чтобы добиться эффективной работы всех составляющих системы, при выборе лазерного принтера нужно учитывать разрешающую способность сканирующего устройства. Наиболее оптимальным соотношением разрешающей способности сканера и лазерного принтера, по всей видимости, будет соотношение 2:1.

При выборе и заказе принтеров необходимо проработать все вопросы, связанные с комплектующими материалами, прикинуть расход красящей ленты, порошка лазерных принтеров, бумаги различного качества и т.д., поскольку комплектующие изделия также заказываются отдельно.

Здесь не приводятся характеристики конкретных принтеров. Для этого существуют каталоги и проспекты. Мы лишь очертили круг вопросов, на которые вы должны дать четкий ответ при выборе принтеров для вашей издательской системы.

# Принтер STAR LC-15

Представляем нашим читателям новейший принтер из серии LC фирмы STAR MICRONICS. Его любезно предоставил нам для испытаний представитель фирмы Кшиштоф Мусял.

Серия LC - это принтеры, конструируемые и продаваемые в соответствии с девизом "Много за мало". Другими словами, это широкие возможности печати, высокое качество, безаварийность и простота обслуживания при такой цене, которая гарантирует фирме доходы благодаря большой продаже.

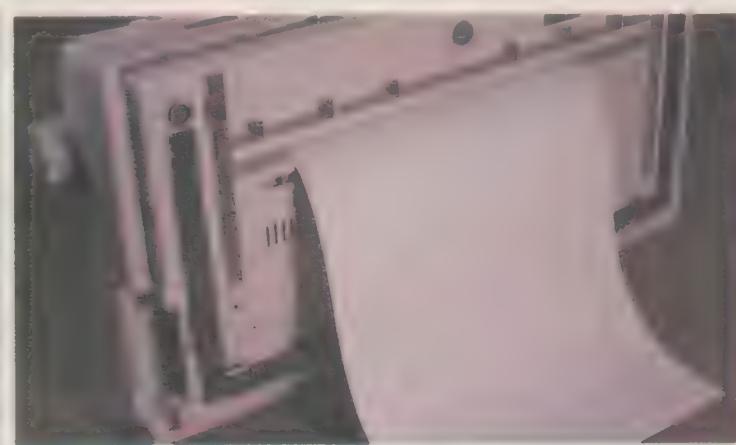
Модель LC-15 представляет собой матричный 9-игольный принтер с валиком 42 см (максимальная ширина печати - 40 см). В одной строке может быть отпечатано от 136 символов типа "rca" до 272 символов типа "condensed elite". Высота символов - 2,2 мм.

Разъем интерфейса помещен в специальную кассету (как в принтерах NL-10), что позволяет быстро поменять интерфейс (такой возможности не было в предыдущих моделях серии LC). Обычно принтер снабжается параллельным интерфейсом Centronics, но можно приобрести и последовательный интерфейс типа RS 232.



Интерфейс имеет буфер ОЗУ емкостью 15 Кбайт, что значительно ускоряет процесс печати. Если пользователь загружает собственный шрифт (download), емкость буфера уменьшается. В таком случае при печати он способен принять 275 байт (символов), а оставшееся пространство будет занято шрифтом пользователя. Можно запрограммировать 255 собственных символов в обоих режимах печати (draft и NLQ), с любыми кодами таблицы ASCII - от 1 до 255 десятично (01 - FF шестнадцатерично). На советский рынок принтеры LC-15 поставляются с РПЗУ, содержащим символы русского алфавита в одном из двух стандартов: КОИ-8 или "Основная кодировка ГОСТа".

Как и в предыдущих моделях серии, принтер позволяет использовать не только отдельные листы бумаги, но и бумажную ленту с перфорацией по краям, причем оба типа бумаги могут быть введены одновременно. Новшеством является дополнительный



вход для бумажной ленты в днище корпуса принтера.

Введение отдельных листов полностью автоматизировано. Лист надо вложить в направляющие и нажать кнопку SET/EJECT PARK. Бумага "занимает" рабочее положение, а принтер звуковым сигналом сообщает о готовности к работе. Автомат введения бумаги работает безотказно, независимо от величины листа.

В принтере LC-15 применена такая же кассета с 13-миллиметровой красящей лентой, как и в модели LC 24-10. На одной ленте можно отпечатать около 6 млн. символов типа draft, что не так уж много для принтера с 42-сантиметровым валиком.

Много внимания конструкторы LC-15 уделили борьбе с шумом. Внутри корпуса выклейен звукоизолирующим материалом. В механизмах принтера нет непосредственно взаимодействующих металлических элементов. Ряд механических элементов выполнен из материалов типа тefлона. Принтер LC-15 является самым тихим из всех печатающих устройств ударного действия, с какими мне приходилось иметь дело.

К сожалению, конструкторы отказались от доступной в серии N функции установки левых и правых полей с помощью клавиш пульта управления. В то же время введена возможность передвижения бумаги вверх и вниз с шагом примерно 0,4 мм.

В модели LC-15 предусмотрена функция измерения длины листа. После автоматического введения листа бумаги принтер можно выключить, а затем вновь включить с нажатой клавишей SET/EJECT PARK. В первой строке будет отпечатано сообщение, что эта строка - первая. Далее бумага перемещается вверх и в последней доступной строке выводится сообщение, что это n-ая строка. Благодаря этой функции можно без дополнительных измерений спланировать печать.

В модели LC-15 расширена по сравнению с предыдущими моделями возможность управления печатью с помощью специальных символьных команд. Команды вводятся в формате (\*)?, где "\*" означает прописную букву, а "?" - цифру. Команды, позволяющие, например, ввести наклонную или жирную печать, подчеркивание, изменить ширину или высоту символов или выбрать шрифт, чрезвычайно удобны при работе с простейшими редакторами или для печати "напрямую", т.е. непосредственно с клавиатуры компьютера. Дополнительно команды принтера могут заменить программу-редактор в таких функциях, как центрирование строки, передвижение ее к левому или правому краю, установка полей и количества строк на странице.

В ходе тестирования принтера LC-15 проверялись его текстовые и графические возможности, описанные в инструкции. Все функции описаны и работают правильно. Контрольные распечатки текстовых и графических файлов дали положительные результаты. Принтер без ошибок работал в стандартах Epson и IBM.

Лучшие результаты дает использование драйверов Epson серии FX и EX, а также BM ProPrinter.

Для проверки скорости печати я применил два метода измерений. Мой собственный метод заключается в распечатке текста определенных размеров, состоящего из всех доступных символов ASCII. Измерение времени распечатки позволяет вычислить скорость работы, которая выражается в числе знаков на единицу времени. В режиме draft мой тест дал результат 110 зн./с, а в режиме NLQ - 24 зн./с. Вторым примененным методом был тест IPS (International Printer Standard - Print Time). Он охватывает печать 10 страниц типичного коммерческого письма в режиме draft, 10 страниц в режиме NLQ, 10 страниц цифровых таблиц и 10 страниц с графиками. Время измеряется компьютером. В завершение теста распечатывается таблица с результатами, которые выглядели следующим образом: 84 зн./с в режиме draft, 24 зн./с в режиме NLQ. Результаты теста IPS и моего метода оказались ниже параметров, приведенных в инструкции. Добавлю, что как в контрольных распечатках, так и в тестах, качество печати было хорошим - на уровне других 9-игольных моделей серий L и N фирмы STAR.

Что касается недостатков, то вызывает опасения прочность конструкции подпорки с направляющими для ввода отдельных листов бумаги. Подпорка шириной почти в 0,5 м держится в корпусе только на двух довольно тонких зацепах. Неосторожность оператора может привести к их повреждению.

Определенные неудобства вызывает также необходимость применения дополнительного привода бумаги с краевой перфорацией, если она вводится через щель в днище. Кроме того, этот механизм блокирует возможность введения листовой бумаги.

В завершение можно сказать, что принтер LC-15 является самым современным изделием из серии LC. Он может применяться в учреждениях и на предприятиях, будучи самым дешевым печатающим устройством в своем классе. Его цена, согласно прейскуранту фирмы STAR MICRONICS, составляет 440 дол.

Перевод Анджея Поплавского

## Достоинства LC-15:

- \* Относительно бесшумная работа
- \* Емкий входной буфер
- \* Кассета со сменными интерфейсами
- \* Чрезвычайно богатые возможности печати
- \* Простота обслуживания
- \* Хорошо написанная инструкция обслуживания

## Недостатки LC-15:

- \* Медленная печать (особенно в режиме NLQ)
- \* Недолговечность красящей ленты
- \* Недостаточная прочность конструкции подпорки с направляющими листовой бумаги
- \* Отсутствие возможности установки ширины полей с помощью клавиш пульта управления

## Основные технические данные

Тип печати	ударный, матричный
Печатающая головка	9-игольная
Режимы печати	draft (черновая), матрица 9x11, NLQ (качественная), матрица 18x23
Шрифты	Courier, Sanserif, Orator
Ширина бумаги	40 см
Число сим. в строке	136 типа pica, 272 типа condensed elite
Разрешение графики	устанавливается программно, от 60 до 240 точек/дюйм (от 2,4 до 9,5 точек/мм соответственно)
Стандартный интерфейс	параллельный, Centronics, 13 Кбайт ОЗУ
Красящая лента	13 мм, в кассете
Габариты	ширина - 59 см, длина - 33 см, высота - 13 см
Вес	8,5 кг

Дополнительное оборудование устройство для автоподачи листовой бумаги, дополнительный привод для бумаги с перфорацией, интерфейс RS 232



## На книжную полку

Болокан Д. Microsoft Word 4.0 Simplified for the IBM PC and Compatibles. Windcrest Books, 1988, ISBN 0-8306-2948-3. В книге Д. Болокана можно выделить три части, предназначенные для различных категорий пользователей. Первая часть книги - хороший вводный курс для начинающих пользователей, не требующий какой-либо предварительной подготовки. Во второй части излагаются сведения, необходимые для более опытных пользователей, а в третьей части описываются возможности, предназначенные для профессионалов в области обработки текстов. Многочисленные иллюстрации (почти на каждой странице) весьма помогают при чтении книги. Однако несколько мешают сведения о предыдущих версиях Word. Тем не менее, книга Д. Болокана - лучший выбор для тех, кто продолжает пользоваться версией Word 4.0.

Побывовольский В.Ю. Текстовый процессор Microsoft Word (руководство по эксплуатации). Версия 4.0. - М.: Совместное предприятие "Параграф", 1989. Можно поздравить СП "Параграф" с тем, что оно опередило центральные издательства и выпустило в свет книгу по Word. Тираж книги - 50 000 экз., что явно мало, но многие читатели не покупают ее из-за цены (6 руб.). Книгу можно назвать справочным пособием по Word, но для обучения она не годится. Возможности Word 4.0 описаны достаточно полно, однако изложение материала очень сухое, отсутствуют иллюстрации и предметный указатель. Не хватает и советов по эффективному применению Word. Впрочем, указанные недостатки, к сожалению, типичны для многих советских книг. Так что поблагодарим автора книги и СП "Параграф" за то, что теперь можно купить в магазине описание Word 4.0 на русском языке.

Beason P.S. Mastering Microsoft Word 5.0. Bantam Books, 1989, ISBN 0-553-34670-9. Книга П. Бисона по своей структуре похожа на описанную выше книгу Д. Болокана. В ней также очень много иллюстраций и примеров, и предназначена она для начинающих пользователей. К достоинствам книги можно отнести многочисленные советы автора о том, как лучше применять Word. Подробно излагаются приемы работы для построения таблиц, оглавлений и предметных указателей, включая иллюстраций. Хорошо структурированное изложение делает чтение книги легким и приятным. В целом эту книгу можно смело рекомендовать нашим читателям. Издательство "Финансы и статистика" намерено предпринять ее перевод на русский язык.

Beason Pamela S. Guild Stephen. Mastering Microsoft Word for Windows. John Wiley & Sons, 1990, ISBN 0-471-50064-X. В книге описывается новый программный продукт фирмы MICROSOFT - Microsoft Word for Windows. Это не адаптация Microsoft Word для работы под управлением системы Microsoft Windows (кстати, такая адаптация и не нужна - Word 5.0 прекрасно работает под Windows), а совершенно другой программный продукт. Word для Windows - издательская система, которая работает в соответствии с принципом WYSIWIG - то, что изображается на экране (в частности, символы различных гарнитур и кеглей, рисунки и т.д.), полностью соответствует тому, что будет выведено на печать. Word для Windows имеет почти все возможности Word 5.0 и много того, чего там нет (ввод формул, удобное оформление таблиц, автоматически вычисляемые поля в таблицах и т.д.), но его пользовательский интерфейс совсем другой. В первых главах книги П. Бисон и С. Гилда описывают новые возможности Word для Windows и соотношения между ним, Windows и MS-DOS. Это очень удобно для тех, кто уже

знаком с Word, однако начинающим пользователям читать данный материал будет весьма трудно. В издательстве "Финансы и статистика" рассматривается возможность перевода этой книги на русский язык.

Beacham Walton. Beacham Deborah. Wordperfect 5.0 Handbook. Bantam Books, 1988, ISBN 0-553-34493-5. Прекрасный учебник по текстовому процессору WordPerfect 5.0, ведущий пользователя от концептуальных представлений к их практическому воплощению. Книга содержит подробное описание команд, меню, приемов работы, создания виртуальных клавиатур. Обсуждаются вопросы работы со шрифтами, принтерами, комбинирования графики с текстом, отдельные свойства, приближающие текстовый процессор к настольной издательской системе, а также создания и использования макросов. Книга, безусловно, будет полезна изучающим WordPerfect и поможет углубить свои знания профессиональным.

Bove Tony. Rhodes Cheryl. Desktop Publishing with Page-Maker for the IBM PC AT, PS/2 and Compatibles. John Wiley & Sons, Inc., 1987, ISBN 0-471-62618-X. В книге описывается полный поэтапный процесс выпуска издания: от планирования работ до получения готового к тиражированию оригинал-макета (camera-ready copy). Подробно описывается и анализируется процесс верстки оригинал-макета в среде Ventura Publisher: компоновка страниц из текстовых и графических блоков; подготовка текстов и иллюстраций. Книга носит ярко выраженное практическое направление, содержит полное описание меню, команд и приемов работы с программой верстки Ventura. Может служить незаменимым подспорьем при подготовке изданий с помощью Ventura Publisher. Книга, возможно, будет издана в переводе в издательстве "Финансы и статистика".

Maker. Издательство "Финансы и статистика" готовит русский перевод этого издания в серии "Новые информационные технологии" в издательском деле".

Thompson Kate H. Randall Peter G. Bennet Steven J. Instant PageMaker (IBM Version 3.0) 20 Ready-to-Run Templates. Simon & Schuster, Inc., 1988, ISBN 0-13-468034-0. В книге кратко описываются этапы подготовки публикаций в среде PageMaker, работа с принтерами и наборным оборудованием, подготовка текстов и иллюстраций. Внимание авторов сконцентрировано на верстке большого числа разнотипных публикаций (подготовка текстов и графики для них) с использованием заранее подготовленных элементов формата (шаблонов) разного уровня (страницы, параграфы и т.д.). Книга снабжена готовыми шаблонами на гибком диске и может быть полезна при повседневной практической работе с программой верстки PageMaker.

McClure Elisabeth. Ventura Publisher: A Creative Approach. Windcrest Books, 1990, ISBN 0-8306-9312-2. Автор описывает поэтапный процесс выпуска издания: от планирования работ до получения готового к тиражированию оригинал-макета (camera-ready copy). Подробно описывается и анализируется процесс верстки оригинал-макета в среде Ventura Publisher: компоновка страниц из текстовых и графических блоков; подготовка текстов и иллюстраций. Книга носит ярко выраженное практическое направление, содержит полное описание меню, команд и приемов работы с программой верстки Ventura. Может служить незаменимым подспорьем при подготовке изданий с помощью Ventura Publisher. Книга, возможно, будет издана в переводе в издательстве "Финансы и статистика".

Составили  
Виктор Фигурнов,  
Михаил Сальников

# Пишем файл CONFIG.SYS

Как можно облегчить работу с компьютером ВМ РС? Ответ прост – для этого надо создать такую среду, в которой пользователю было бы удобно работать. Лучше всего, если эта среда устанавливается автоматически при начальной загрузке MS-DOS. Для того чтобы пользователи могли создать удобную для них рабочую среду, операционная система MS-DOS предоставляет различные средства конфигурирования. В настоящей статье описываются эти средства и даются советы по их применению.

## Копирование файлов с жесткого диска

Как правило, фирмы-поставщики компьютеров поставляют компьютеры с жестким диском, на котором записаны операционная система MS-DOS и не большое число других программ (например, драйверы клавиатуры и принтера или транслятор языка BASIC). Все остальные программы пользователь должен установить на жесткий диск сам.

Перед тем как записывать на жесткий диск новые программы, желательно скопировать с него операционную систему MS-DOS и другие программы на дискеты. Эти дискеты можно будет использовать при повреждении файлов операционной системы на жестком диске и в тех случаях, когда компьютер не загружается с жесткого диска (например, при повреждении информации в постоянной памяти компьютера).

Перед копированием файлов с жесткого диска на дискеты целесообразно включить режим проверки записи на диск. Это позволит избежать создания неправильных копий файлов на дискетах. Режим проверки записи на диск включается командой VERIFY ON. После окончания копирования файлов на дискеты этот режим можно выключить командой VERIFY OFF.

Для того чтобы скопировать основные файлы операционной системы, надо ввести следующие команды (мы будем предполагать, что жесткий диск обозначается С:).

С:

SYS A:

COPY COMMAND.COM A:

Если ваш компьютер имеет дисковод для дискет емкостью 1,2 Мбайта (размером 3,25 дюйма) или 1,4 Мбайта (размером 3,5 дюйма), то лучше скопировать все файлы с жесткого диска на одну дискету. Копирование выполняется с помощью команды

XCOPY C:\\*.\* A: /S /E

Следует заметить, что иногда в поставленной вместе с компьютером версии операционной системы MS-DOS программа XCOPY отсутствует. В этом случае файлы из каждого каталога на жестком дис-

ке можно скопировать командой COPY в соответствующий каталог на дискете.

На дискету с операционной системой MS-DOS целесообразно также скопировать часто используемые программы для обслуживания файловой системы на диске (например, NDD, DT и QU из комплекса программ Norton Utilities), а также резидентную программу для защиты от вируса (например ANTI4US). Если вы привыкли работать с какими-то программами-оболочками типа программы Norton Commander, то их также следует скопировать на эту дискету.

Если ваш компьютер не имеет дисководов для дискет большой емкости, то файлы с жесткого диска не поместятся на одну дискету. Поэтому следует записать на одну дискету наиболее часто используемые программы операционной системы MS-DOS, в частности программы FORMAT, LABEL, SYS, CHKDSK, а также программы, указанные выше. Кроме того, на эту дискету надо обязательно записать программу для установки параметров конфигурации компьютера (эта программа может называться SETUP, SETUP1, AT-SETUP и т.п.), если такая программа имеется в комплекте программ, поставляемых с компьютером. Остальные файлы, находящиеся на жестком диске, скопируйте на другие дискеты. Полезно на каждую из этих дискет поместить и командный процессор MS-DOS – файл COMMAND.COM, чтобы при работе с ними не выдавалось сообщение, требующее вставить дискету с файлом COMMAND.COM.

После того как все файлы с жесткого диска будут скопированы на дискеты, следует заклеить на этих дискетах прорезь защиты от записи, чтобы скопированные файлы не могли быть случайно изменены или испорчены. Целесообразно также сделать копии этих дискет, и хранить вторые экземпляры дискет отдельно от первых, чтобы уменьшить вероятность порчи всех копий файлов при несчастном случае.

## Разбиение жесткого диска

После копирования файлов с жесткого диска на дискеты можно при необходимости разбить жесткий диск на несколько частей – логических дисков. Такое разбиение производится специальными программами, например, программой Disk Manager. При разбиении жесткого диска на несколько логических дисков на компьютере можно одновременно использовать две операционные системы, например MS-DOS и UNIX (XENIX). Каждый логический диск, доступный операционной системе MS-DOS, имеет собственное обозначение (C, D, E и т.д.), и для пользователя работа с логическим диском ничем не отличается от работы с каким-либо другим диском.

Разбиение жесткого диска можно сделать и позднее, но тогда придется сохранять все файлы с жесткого диска на дискетах, а потом копировать их обратно на жесткий диск.

Необходимость в разбиении жесткого диска на части может быть вызвана следующими причинами.

1. При разбиении жесткого диска на логические диски можно сделать любой логический диск (кроме того, с которого загружается операционная система MS-DOS) защищенным от записи, причем защиту от записи можно установить и снять в любой момент времени. В частности, можно записать на логический диск программы и данные, которые не требуется каким-либо образом изменять, а затем установить на этом диске защиту от записи. Это позволяет обеспечить сохранность и защиту от вируса программ и данных, которые вы не изменяете.





2. С помощью разбиения жесткого диска на части на компьютере можно одновременно использовать несколько операционных систем.

3. Если на одном компьютере должны работать несколько пользователей или групп пользователей, то может оказаться целесообразным разделить его жесткий диск на несколько частей, чтобы каждому пользователю или каждой группе пользователей был выделен свой логический диск для размещения файлов. Такой подход, как правило, приводит к значительному уменьшению числа конфликтов, связанных с недостатком места на диске или уничтожением "чужих" файлов.

4. Если на компьютере установлен жесткий диск емкостью более 32 Мбайт или "нестандартный" жесткий диск, т.е. диск типа, не предусмотренного в операционной системе MS-DOS, то при непосредственном применении такого диска возможно существенное недоиспользование его емкости. Один из способов избежать этого - разбиение диска на несколько частей.

Если на жестком диске имеется несколько логических дисков, доступных операционной системе MS-DOS, то с одного из них производится загрузка MS-DOS. Этот логический диск всегда доступен и для чтения, и для записи. Остальные логические диски могут быть доступны для чтения и записи или только для чтения. Для того чтобы с этими дисками можно было работать, необходимо включить в файл CONFIG.SYS строку

device = имя-файла-драйвера-логического-диска

Например,

device=C:\EXE\SYS\HARDRIVE.SYS.

Если разбиение жесткого диска на несколько частей выполняется для того, чтобы на компьютере можно было использовать совместно с MS-DOS какую-то другую операционную систему, то для выполнения разбиения следует воспользоваться программами, указанными в руководстве по установке на компьютер IBM PC этой операционной системы.

## Размещение файлов на жестком диске

При использовании дискет, как правило, не приходится задумываться о том, каким образом следует разместить на них файлы. Чаще всего все файлы на диске записываются в корневой каталог, иногда создаются несколько каталогов для хранения отдельных групп файлов. Более сложные структуры каталогов на дискетах применяются редко, так как на дискетах не помещается такое количество файлов, при котором простейшие структуры дерева каталогов являлись бы недостаточными.

При работе с жестким диском ситуация совершенно другая. На жестком диске можно поместить сотни и тысячи файлов, и без продуманной схемы их размещения разбираться в них было бы крайне трудно. Кроме того, рассредоточивать файлы на жестком диске по различным каталогам полезно еще и потому, что при размещении в одном каталоге большого количества файлов MS-DOS будет затрачивать много времени на поиск файлов в этом каталоге. Замедление доступа к файлам в каталоге начинает ощущаться при 50-100 файлах, а при двухстах файлах в каталоге доступ к ним занимает несколько секунд.

Следует позаботиться, чтобы в корневом каталоге диска не было большого количества файлов и подкаталогов. Это не только ускоряет работу с диском, но и облегчает ориентировку в файловой системе на дис-

ке. Полезно руководствоваться таким правилом: в корневом каталоге жесткого диска, с которого загружается операционная система MS-DOS, должны находиться только те файлы, которые не могут находиться в каком-либо другом месте. Такими файлами являются файлы операционной системы MSDOS.SYS и IO.SYS (или IBMDS.COM и IBMIO.COM), файлы AUTOEXEC.BAT и CONFIG.SYS. При использовании программ Norton Commander (версии ниже 3.0) и NCD в корневом каталоге диска должны находиться также файлы NC.MNU и TREEINFO.NCD.

Кроме этих файлов в корневом каталоге диска должны находиться подкаталоги для используемых пакетов программ, для исполнимых файлов общего назначения, для пользователей или групп пользователей, а также определенных видов работ и т.д. Следует стремиться, чтобы число подкаталогов, находящихся непосредственно в корневом каталоге, было не слишком велико.

Некоторые имена каталогов являются широко распространенными, например, EXE - исполнимые программы и командные файлы общего назначения, DOC - документы, WORK - каталог для случайных работ и пользователей, VENTURA - издательская система Ventura Publisher, TC - транслятор Turbo C и т.д. В этих каталогах можно размещать подкаталоги для определенных видов файлов, например, EXEDOS - программы, входящие в операционную систему MS-DOS; EXEVSYS - драйверы, указываемые их в предложении device файла CONFIG.SYS; TC\INCLUDE - файлы заголовков, включаемые с помощью оператора include в транслируемые тексты программ на Turbo C; TP\DOC - документы по транслятору Turbo Pascal и т.д.

Если число подкаталогов в корневом каталоге получается слишком большим, то можно сделать несколько каталогов аналогичного назначения подкаталогами одного каталога.

В заключение приведем еще несколько рекомендаций по размещению файлов на жестком диске.

1. Все файлы, относящиеся к какому-то одному комплексу (пакету программ), надо размещать в одном каталоге и его подкаталогах.

2. Как правило, не следует помещать в одном каталоге файлы, относящиеся к разным комплексам (пакетам программ).

3. Не нужно помещать в одном каталоге файлы, которые вы используете, но не изменяете, и файлы, которые вы изменяете или можете изменить. Это упростит процесс архивации файлов.

4. Все исполнимые файлы и командные файлы общего назначения желательно помещать в одном каталоге и нескольких (не более двух-трех) его подкаталогах. Эти каталоги следует указать в команде MS-DOS Path при написании файла AUTOEXEC.BAT.

5. Если вы захотите выполнить на компьютере какой-то новый вид работ, то лучше создать новый каталог для проведения этих работ и размещения соответствующих файлов.

## Написание файла CONFIG.SYS

В корневом каталоге жесткого диска, с которого загружается операционная система, может находиться файл CONFIG.SYS, задающий параметры операционной системы MS-DOS, а также указывающий, какие программы, расширяющие возможности операционной системы, необходимо загружать в оперативную память (эти программы называются драйверами устройств, хотя они не обязаны управлять работой каких-либо устройств).

Файл CONFIG.SYS должен представлять собой текстовый (ASCII) файл. Каждая строка этого файла имеет вид:

имя-команды = значение.

Опишем наиболее часто используемые команды файла CONFIG.SYS.

Break = on - установить режим проверки нажатия [Ctrl-Break] при операциях ввода-вывода с диском. Это позволит при необходимости прерывать выполнение программ.

Buffers - число-буферов - установка числа буферов для операций ввода-вывода с диском. Для IBM PC без жесткого диска рекомендуется использовать 4-5 буферов, для IBM PC XT с жестким диском емкостью 10-20 Мбайт - 15-20 буферов, для IBM PC AT с жестким диском емкостью 20-40 Мбайт - 30-40 буферов.

Country = 049 (для MS-DOS версий до 3.2), Country = 049,437,полное-имя-файла-COUNTRY.SYS (для версий MS-DOS, начиная с 3.3) - установка удобного формата выдачи информации о дате и времени.

Files = 20 - установка максимального числа одновременно открытых файлов. Для выполнения многих программ необходимо, чтобы значение параметра FILES было не менее 20.

Shell = COMMAND.COM /E:число-байтов /P - увеличение размера области памяти, в которой хранятся переменные окружения. Число байтов задает размер этой области. Если размер области памяти для хранения переменных окружения недостаточен, то MS-DOS выдает сообщение "Out of environment space".

Device - имя-файла-драйвера [параметры] - установка драйвера устройства. Например, если файлы драйверов находятся в каталоге C:\EXE\SYS, то:

device = C:\EXE\SYS\ANSI.SYS - установка драйвера ANSI.SYS, расширяющего возможности по выводу на экран и позволяющего переопределять значения клавиш на клавиатуре;

device = C:\EXE\SYS\VDISK.SYS размер-диска [/E] - установка "электронного" диска. Размер диска задается в килобайтах. Параметр /E указывает, что "электронный" диск должен размещаться в расширенной памяти процессора Intel-80286 или 80386.

Более подробно эти и другие команды файла CONFIG.SYS изложены в руководстве по MS-DOS.

В следующем выпуске "Компьютера" читайте о составлении файла AUTOEXEC.BAT.

### Пример файла CONFIG.SYS

```
Break = on
Files = 20
Buffers = 16
Shell = COMMAND.COM /E:512 /P
Device = C:\EXE\SYS\ANSI.SYS
Device = C:\EXE\SYS\HARD_DRIVE.SYS
Device = C:\EXE\SYS\MOUSE.SYS
Device = C:\EXE\SYS\VDISK.SYS 384 /E
```

С \ Компьютер на работе \

© Ян Рожнович

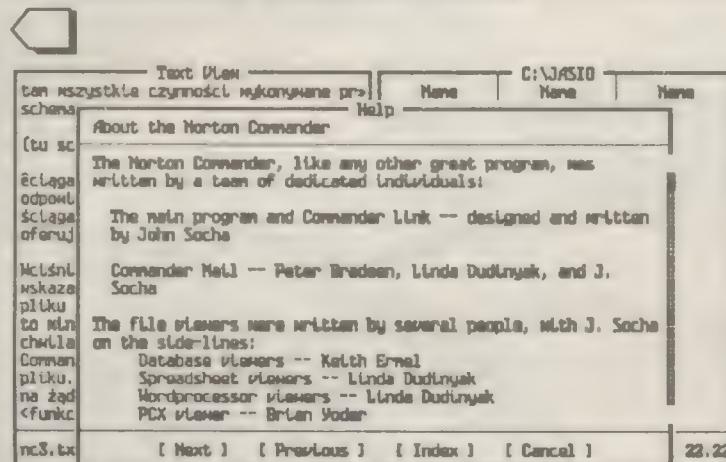
# Norton Commander ver. 3.0

О программе Norton Commander наши читатели могли прочитать в предыдущем выпуске "Компьютера". Относительно недавно, в октябре 1989 г. Питер Нортон выпустил новую версию своей программы-оболочки под номером 3.0. Она значительно богаче своей предшественницы - версии 2.5, что вполне оправдывает присвоение ей нового порядкового номера (согласно не-писанной договоренности авторов программного обеспечения изменение первой цифры порядкового номера версии равнозначно существенному изменению или дополнению программы).

Norton Commander 3.0 занимает две 360-Кбайтовые диски либо одну 720-Кбайтовую (или 1,2-Мбайтovую). На них размещены инсталляционная программа, автоматически обнаруживающая на диске, где устанавливается Norton Commander, присутствие предыдущей версии оболочки и (по желанию пользователя) заменяющая ее новой, а также собственно файлы Norton Commander, часть которых хранится в упакованном виде. Устанавливаясь программа осуществляет также пересылку файла NC.MNU из корневого каталога в каталог, где хранится Norton Commander. В новой версии все рабочие файлы программы (за исключением TREEINFO.NCD и локальных NC.MNU) находятся в одном каталоге.

Norton Commander 3.0 насчитывает 15 EXE-файлов, 5 из которых имеют в своем имени слово "view". Они действительно обеспечивают просмотр различных типов файлов. Небольшая резидентная часть Norton Commander 3.0 с именем NC.EXE соответствует файлу NCSMALL.EXE версии 2x и насчитывает всего 3 Кбайта. NC.EXE в новой версии адекватен NCMAIN.EXE.

После запуска версии 3.0 внешний вид экрана почти такой же, как в версии 2.x. Единственное, чисто косметическое изменение - в подсказке в нижней части экрана вместо слова USER стоит MENU (F2), а слово MENU заменено на PullOn (F9). Однако стоит только нажать клавишу Alt, и сразу появляется другая подсказка, определяющая новое значение функциональных клавиш, если их нажимать одновременно с Alt. Самое существенное отличие новой версии от старых обнаруживается, если нажать F1 (HELP). Вместо краткого описания значения отдельных клавиш на экране появляется обширная, полностью удовлетворяющая любопытство пользователя система контекстной шпаргалки. В ней - подробное описание всех действий, выполняемых программой Norton Commander 3.0, тщательно отработанные примеры и даже схемы, иллюстрирующие самые подробные детали.

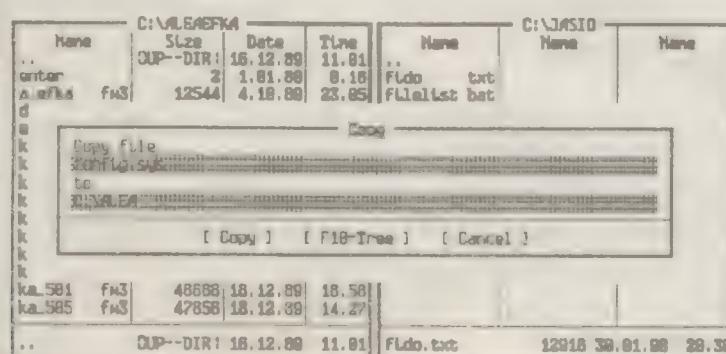


В новой версии возможности просмотра различных типов файлов (F3) гораздо шире. Но их загрузка — процесс ощутимо более продолжительный (острики утверждают, что во всем, как обычно, виновата женщина, написавшая этот фрагмент программы). На самом деле причина заключается в том, что в новой версии для просмотра файлов Norton Commander загружает одну из внешних программ. Возможности просмотра при этом значительно больше. По требованию можно отформатировать до ширины экрана длинные строки (функция Wrap — F3). Содержание просматриваемого файла можно вывести на экран не только в виде кодов ASCII, но и в виде шестнадцатеричных кодов (функция Hex или ASCII — F4). Это очень полезно при просмотре нетекстовых файлов. В зависимости от формата файла можно также изменить по ходу дела программу его просмотра, что существенно при работе с различными текстовыми редакторами (функция Viewer — F8).

00948	78	8F	77	68	61	84	81	28	63	6F	20	86	61	84	60	86	polwida co najmniej jedna kusznica w tancu strona tak stw. Jednak F73 guska naszta dos konieca, to nikt nie zna. Co najmniej jedna kusznica na koniec?
00958	69	85	64	28	64	65	64	28	61	28	77	79	96	77	89	85	
00963	74	86	61	66	61	5	68		3	60	F	73	77	82	87		
00968	73	74	75	2E	68	67	81	77	68	72	73	77	82	87	82		
00972	73	74	75	2E	73	74	75	73	74	75	76	77	78	79	78		
00978	65	8F	65	81	92	65	8F	65	81	92	65	8F	65	81	92		
00981	73	74	75	2E	73	74	75	73	74	75	76	77	78	79	78		
00983	61	77	64	65	28	61	77	64	65	28	61	77	64	65	28		
00988	72	75	64	65	28	62	75	64	65	28	62	75	64	65	28		
00991	57	6D	6A	80	68	62	6C	61	77	68	62	6C	61	77	68		
00992	62	6C	61	77	68	62	6C	61	77	68	62	6C	61	77	68		
00995	66	73	77	83	68	66	73	77	83	68	66	73	77	83	68		
00998	61	61	63	63	28	61	61	63	63	28	61	61	63	63	28		
01001	78	8C	66	68	75	60	84	77	73	5B	61	7A	61	66	65	77	
01003	6F	28	70	72	74	65	74	70	72	68	75	72	73	6F	72	28	

В Norton Commander 3.0 сохранена возможность просмотра файлов формата dBase и Lotus 1-2-3. Новинкой является программа просмотра графических файлов формата PCX. Авторы, конечно, не забыли, что "consuetudo altera natura est" (в переводе с латинского — "привычка — вторая натура (человека)"): нажатие Alt-F3 приводит в действие программу просмотра файлов из старой версии.

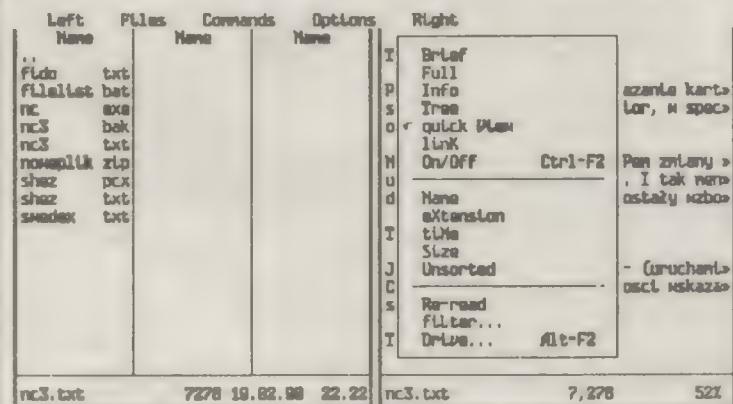
Небольшие, но весьма удобные для работы изменения введены в операциях COPY и MOVE (F5 и F6 или Shift-F5 и Shift-F6).



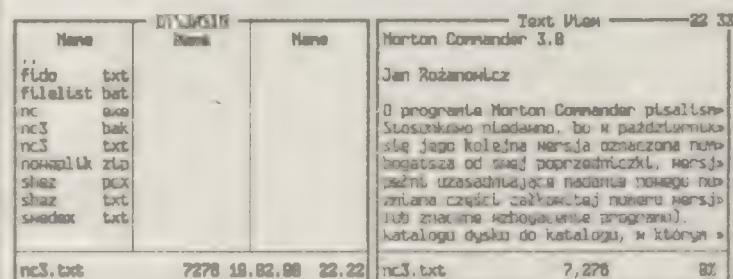
Нажатие клавиши F10 обеспечивает возможность выбора каталога, в который необходимо переместить или откопировать указанный файл. Дерево каталогов появляется в специальной панели.

Слегка изменена операция, приводимая в действие комбинацией клавиш Ctrl-R (просмотр содержимого диска или каталога из текущей панели). Раньше для просмотра требовалось дополнительное подтверждение выбора дисковода. В новой версии — нет.

Наиболее значительно, кроме подсказки, изменено действие клавиши F9: меню режимов работы левой и правой панели дополнено тремя новыми опциями.

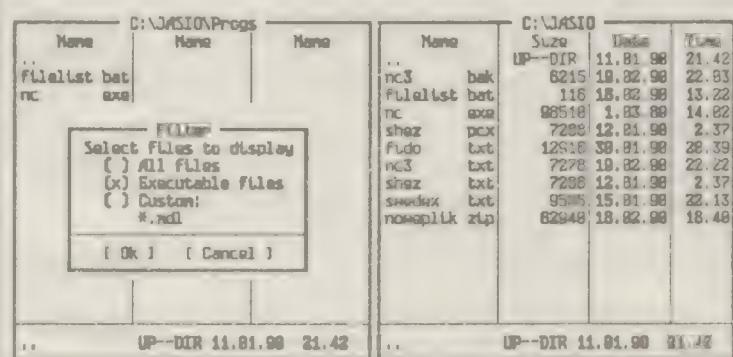


Одна из них — quick View (приводится в действие также нажатием Ctrl-Q) обеспечивает просмотр в соседнем окошке содержимого файла, указанного курсором.



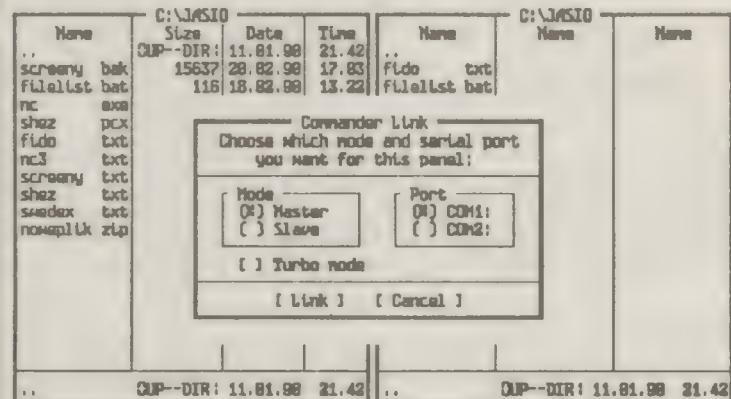
В зависимости от типа файла (расширения его имени) загружается соответствующая программа просмотра. С помощью клавиши F3 панель увеличивается до размеров всего экрана.

Вторая новая опция — Filter позволяет задать шаблон для имен файлов, которые появляются в правой или левой панели. Так, Norton Commander 3.0 может показать все файлы, находящиеся на данном диске или в каталоге, файлы с расширением \*.BAT, \*.COM, \*.EXE, или файлы, соответствующие установленному шаблону.

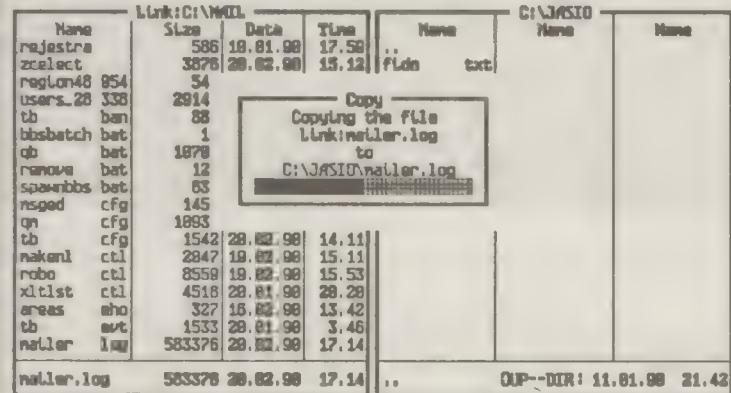


Самое интересное третье новшество. Это канал связи (LINK), который можно использовать, если соединить две микроЭВМ через последовательный интерфейс (COM1 или COM2). Вы устанавливаете на обеих маши-

нах опцию LINK, заранее определив, какая из них главная (MASTER), а какая - подчиненная (SLAVE), и Norton Commander 3.0 обеспечит с главной ЭВМ доступ к дискам подчиненной. Содержимое диска подчиненной микроЭВМ будет находиться в одной из панелей экрана главной.

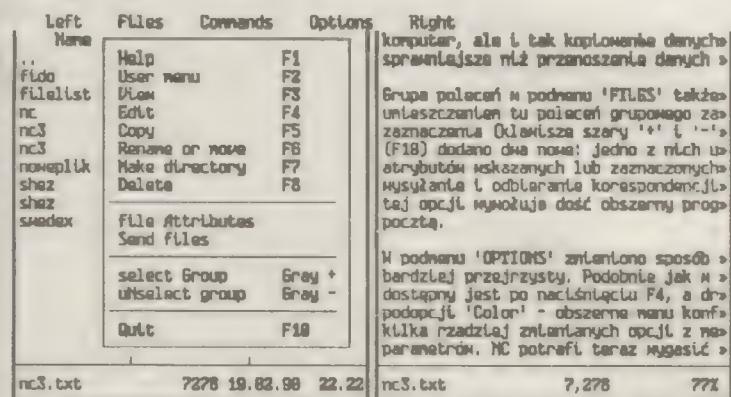


После этого можно работать с файлами, находящимися на жестком диске подчиненной машины, например, копируя их на жесткий диск главной.

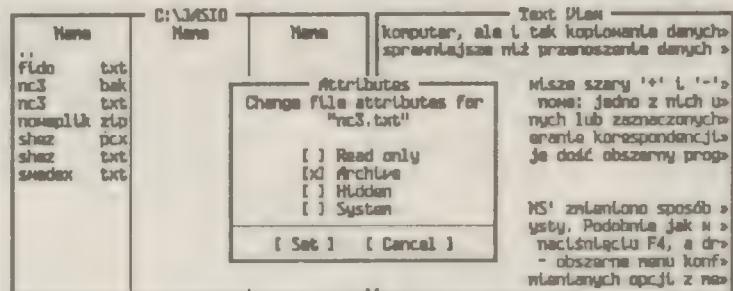


Передача данных осуществляется со скоростью 13 Кбит/с. И хотя это намного медленнее, чем скорость передачи данных с жесткого диска в ЭВМ, однако копирование файлов с одного компьютера на другой с помощью этой опции более удобно, чем при посредничестве дискет.

Разработчики NC 3.0 увеличили также число команд в меню FILES.

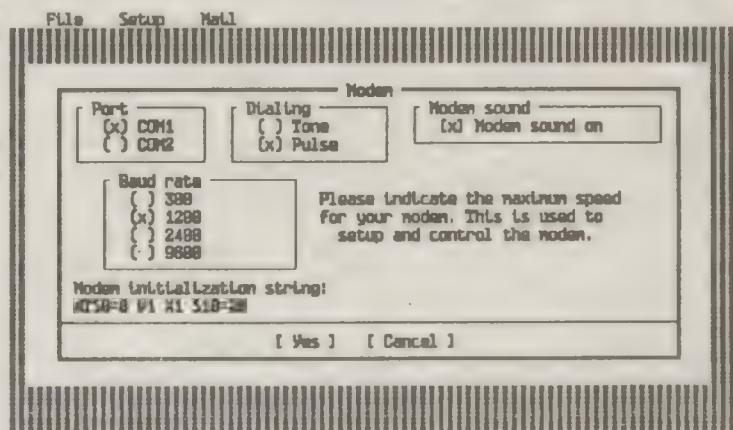


Кроме команд выбора и отмены выбора группы файлов (в старых версиях они приводились в действие только нажатием клавиш [+] и [-] на функциональной клавиатуре), а также выхода из Norton Commander (F10), мы видим здесь еще две команды. Одна обеспечивает изменение всех атрибутов указанных или выбранных файлов.



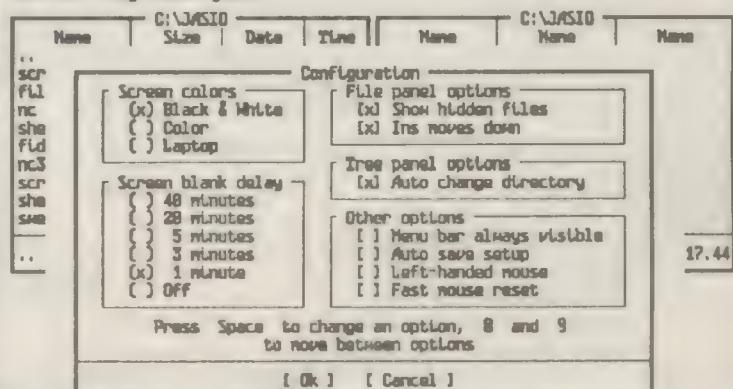
nc3.txt 7278 19.02.98 22.22 nc3.txt 7,278 771

С помощью второй можно получать и отправлять через подключенный к компьютеру модем электронную почту, но только по распространенной в США системе MCI. Выбор этой команды приводит в действие большую программу, управляющую обменом почты.



В меню OPTIONS теперь более удобно указывается, каким внешним текстовым процессором вы будете пользоваться при редактировании файла. Как и раньше, встроенным редактором можно пользоваться, нажав клавишу F4, а указанным внешним - нажав Alt-F4.

И, наконец, последняя новинка, которой нас обрадовал Питер Нортон и его сотрудники - подменю CONFIGURATION. Сюда были перемещены некоторые более редко применявшиеся опции из меню OPTIONS, все старое подменю COLOR, а также введены некоторые новые параметры.



Теперь Commander умеет "гасить" экран по истечении заданного времени. С его помощью можно вывести на одну панель дерево каталогов, а на другую - содержимое данного каталога, заставить программу автоматически "запоминать" setup после каждого его изменения (и тогда при включении компьютера Commander будет "просыпаться" в таком же состоянии, в каком он находился в момент отключения машины). Можно все. Даже переключить мышь для работы левой рукой, если вы левша. Питер Нортон позаботился и об этом.

Перевод Халины Мадейчик

# Антивирусные программы

"Вирус под каждой кроватью" - так была озаглавлена статья в журнале PC-Magazine, посвященная компьютерным вирусам. В этом заголовке отражается паническое состояние одних пользователей ПК и беспечность других. К какой группе присоединиться? Безусловно, не стоит сразу браться за форматирование жесткого диска с полным уничтожением его содержимого при виде прыгающего по экрану дисплея мячика или бегающего "насекомого". С другой стороны, игнорировать проблему тоже не стоит: надо помнить, что одну пятую из примерно 60 известных ныне вирусов написали программисты с террористическими наклонностями. Влияние их "детищ" на ваши файлы лишь незначительно отличается от влияния подложенной рядом с компьютером бомбы.

Что же в такой ситуации делать, как себя обезопасить? Прежде всего надо так организовать работу на компьютере, чтобы опасность заражения свести к минимуму (о принципах "компьютерной гигиены" мы писали в первом выпуске нашего сборника в статье "Десять антивирусных заповедей". - Примеч. ред.). Но полностью ликвидировать ее практически невозможно. Во всяком случае это не легче, чем выработать у себя иммунитет к насморку. Итак, давайте заблаговременно познакомимся с доступными лекарствами: специализированных антивирусных программ возникло уже довольно много.

В принципе существуют три метода программной защиты от вирусов:

- \* специальные программы, которые присоединяются к операционной системе с целью слежения за активностью запускаемых на ЭВМ программ. Если появляются подозрительные симптомы, они блокируют работу данной программы, предотвращают пользователя и ждут его решения. Иногда такие программы-наблюдатели поддерживаются специальными расширяющими картами, которые физически блокируют доступ к программным ресурсам;

- \* программы, вычисляющие контрольные суммы всех доступных на дисках программ и записывающие их в специальный файл. При очередной загрузке они проверяют, не подверглась ли какая-либо программа модификации. В случае положительного результата проверки они предупреждают пользователя. В упрощенном, зато более быстродействующем варианте контролируются единственно длина файла и дата его последней модификации. Это, конечно, исключает выявление вируса, не изменяющего эти параметры;

- \* программы, которые "охотятся" за определенными типами вирусов, чаще всего быстродействующие (хотя бывают и исключения), способные уничтожить "противника" без повреждения программы или диска. Некоторые программы такого типа проверяют, есть ли на диске эффекты разрушительной деятельности вируса.

Я не буду сравнивать здесь достоинства и недостатки разных методов и остановлюсь на рассмотрении первой из указанных групп программ - программ-мониторов, главным свойством которых является то, что они работают вместе с операционной системой и наблюдают за действием других программ. Чтобы хорошо понять идею, заложенную в основу работы этих программ, начнем с механизма функционирования вирусов.

Компьютерные вирусы - это небольшие программы, которые сами не выполняются, а присоединяются к программам-утилитам либо же внедряются на дискету или винчестер и выполняются в момент запуска программы-носителя или загрузки ОС с зараженного диска.

Программа-вирус функционирует в два этапа. Первый этап - это размножение вируса, т.е. код вируса внедряется в различные места вашей системы; куда именно - это зависит от типа вируса. Например:

- \* в разного рода выполняемые файлы (как в классические программы типа EXE или COM, так и в различные программы-оболочки, системные файлы и т.д.);

- \* в загрузочный сектор дискеты, в таблицу разделов (partitions) жесткого диска, а также в его секторы, обозначенные как поврежденные, на дополнительные дорожки, в дополнительные секторы или даже во временно свободные секторы корневого каталога.

Некоторые вирусы живут и размножаются в операционной системе, другие - в зараженных программах.

Наличие первого этапа, т.е. размножения, отличает вирусы от давно известных "тロjanских коней" - программ, которые под прикрытием нормальной работы, нередко полезной или развлекательной, выполняют функции, типичные для второго этапа деятельности вирусов.

Второй этап, который можно назвать фазой разрушения, чаще всего наступает при определенных условиях или же спустя некоторое время, достаточное для того, чтобы вирус широко распространился перед моментом "рассекречивания", к которому эта фаза ведет. То, что делает вирус на этапе разрушения, является производной знаний, большой фантазии и степени недоброжелательности его создателя. Например, вирус может:

- \* уничтожить на диске системные файлы, что блокирует возможность чтения его содержимого;

- \* отформатировать всю дискету или некоторые дорожки;

- \* уничтожить случайно выбранные секторы;

- \* изменить параметры винчестера или дисковода в памяти конфигурации, в результате чего компьютер "не видит" диск или дискету;

- \* привести к другим осложнениям (снижение скорости работы компьютера, деформация выводимой на дисплей информации или даже попытка физического повреждения оборудования - дисплея, принтера, дисковода или винчестера).

Итак, главным объектом атаки (а на этапе размножения единственным) являются диски вашего компьютера. Это наводит на мысль, что можно попытаться заблокировать атаку, непрерывно контролируя обращения к дисководам и отфильтровывая те, которые выглядят подозрительно. Именно эта идея заложена в основу работы рассматриваемых здесь антивирусных программ.

Давайте подумаем, что должна уметь такая программа?

1. Чтобы заблокировать присоединение кода вируса к ОС, она должна предостерегать в случае по-

попытки неизвестной программы стать резидентной.

2. Чтобы пресечь возможность размножения вирусов, программа должна противодействовать такой модификации исполняемых файлов, которая не является частью нормальной работы системы и в то же время необходима для заражения файла вирусом.

3. Попытки непосредственного (в обход системы файлов DOS) доступа к дискам, особенно к их системным областям (загрузочный сектор, таблица разделов жесткого диска, каталоги или таблицы размещения файлов), могут быть результатом размножения вируса или элементом этапа разрушения. Пользователь должен знать об этом заранее. Стоит вести своего рода дневник модификаций винчестера, что в случае повреждения файлов облегчит поиск виновника.

4. Программа должна регулярно проверять содержимое памяти конфигурации компьютера PC/AT, что позволит избежать трудностей с распознаванием дискеты после рестарта ОС. Необходимой является также возможность восстановления первичных параметров конфигурации при выявлении незапланированных изменений.

5. Программа-монитор должна сравнивать контрольные суммы запускаемой программы с вычисленными ранее и записанными в отдельный файл величинами. Это позволит выявить изменения в программе, прежде чем она будет выполнена.

Итак, ясно, какими свойствами должны обладать антивирусные программы рассматриваемой группы, чтобы предупреждать разрушительную деятельность вирусов. (Есть ряд программ, которые позволяют лишь констатировать то, что уже случилось, и уничтожить вирус.)

Остается еще ответить на вопрос, насколько "прочны" антивирусные программы-мониторы?

Ответ был бы прост, если можно было бы допустить, что вирусы - это "хорошо написанные программы", т.е. такие, которые любые действия осуществляют посредством функций DOS или в крайнем случае BIOS, а также что антивирусная программа запускается на компьютере, память которого свободна от вирусов (иначе говоря, к операционной системе вирус не "прицепил" дополнительные процедуры в виде резидентных программ).

К сожалению, по некоторым причинам из таких предпосылок исходить нельзя. Рассмотрим эти причины.

1. Довольно многочисленна группа вирусов загрузочного сектора и таблицы разделов жесткого диска, которые активизируются в памяти перед загрузкой ОС, т.е. перед загрузкой антивирусной программы. Со стороны многих антивирусных программ действие таких вирусов не контролируется. Случается также, что вирус нельзя обнаружить при чтении сектора, в котором он "поселился", - сектор выглядит таким же, каким был до заражения.

2. Существует возможность присоединения фрагмента кода к системе путем манипулирования величинами в заголовках блоков памяти без использования функций DOS. Это можно выявить, периодически проверяя содержимое заголовков, но пока я не встречал программ (ни коммерческих, ни общедоступных), которые это делают.

3. Начинают появляться вирусы, которые в ПЗУ находят адреса начала процедур обслуживания дисководов и выполняют непосредственный переход на найденный адрес. Это невозможно выявить чисто программным путем. Но все известные мне вирусы такого типа выполняют одну из операций через DOS, благодаря чему они могут обнаруживаться программами-мониторами. Тем не менее возможно создание вируса, который все свои функции выполнит путем описанных выше непосредственных переходов. В та-

ком случае возникает вопрос: стоит ли вообще пользоваться программами-мониторами? Безусловно, стоит, и причин тому несколько:

\* вирусов, заражающих программы, намного больше, чем вирусов, заражающих загрузочные секторы;

\* заражение загрузочного сектора легче предотвратить организационными методами;

\* не все авторы вирусов отличаются изобретательностью, поэтому большинство их "шедевров" можно выявить с помощью программ-мониторов;

\* метод, состоящий в проверке контрольной суммы программы перед ее выполнением, чрезвычайно эффективен по отношению к вирусам, заражающим программы.

Итак, обсудив все "за" и "против", можно прийти к следующему выводу: программы-мониторы стоит применять в качестве антивирусной защиты, однако следует помнить, что они не полностью исключают возможность заражения.

Для того чтобы повысить безопасность, необходимо помнить об упомянутой "антивирусной гигиене" и организационных мерах. В ситуации, когда опасность заражения системы резко возрастает (например, при работе на ней многочисленных пользователей), стоит также время от времени использовать антивирусную программу, проверяющую контрольные суммы, характерные байты или же выявляющую вирусы какими-либо другими методами (лучше всего после загрузки системы со специально подготовленной для этой цели дискеты с закрытым вырезом маркера защиты).

Теперь я предлагаю краткий обзор существующих в начале 1990 г. программ-мониторов и их сравнение с описанным выше "идеалом". Рассматриваться будут только программы типа shareware и общедоступные программы (так как не думаю, чтобы кто-либо из читателей собирался израсходовать на коммерческий продукт такого типа сумму, нередко превышающую 100 дол.).

#### DPROTECT.COM версия 1.01, автор Дж. М. Уонг

Примитивная программа, написанная в период, когда главной опасностью были "тロянские кони". Она блокирует на указанных дисководах все разрушающие операции (запись, форматирование), проводимые посредством дискового BIOS. В случае выявления запрещенной операции программа требует рестарта системы. Для снятия защиты необходима перезагрузка системы. Программа может пригодиться лишь при проверке работы новых программ неизвестного происхождения.

#### HDSENTRY.COM версия 1.01, автор Эндрю М. Фрайд

Программа напоминает DPROTECT. Ее главное отличие - в возможности продолжения работы после выявления запрещенной операции записи или форматирования. Обнаружив попытку такой операции, программа сообщает о ней, после чего, заблокировав ее, продолжает наблюдение. Дискеты защищаются также от разрушающих операций, использующих прерывание 16h. Программа полезна только при проверке новых программ.

#### BOMBSQUAD.COM версия 1.2, автор Энди Хопкинс

По своей работе программа также напоминает DPROTECT, но она гораздо более "дружественна". При выявлении запрещенной операции можно разрешить ее выполнение. Конфигурация программы изменяется без повторной загрузки системы. Полезность программы - на уровне DPROTECT.





### TRAPDISK.COM версия 1.0

Вариант программы BOMBSQUAD, разработан неизвестным автором, который, не считая косметических изменений, добавил обслуживание дополнительных функций дискового BIOS.

### VIRBLOCK.EXE версия 2.1, автор Михель Фитц

Работа известного автора антивирусных программ из Вены. Программа блокирует запись в файлы типа COM и EXE. В случае выявления нелегальной операции - обращается с запросом к оператору, а затем действует в зависимости от ответа. Выполняется только часть функций, которых следовало бы ожидать от программы-монитора, однако за неимением другого она может использоваться против некоторых типов вирусов.

### ANTIUS2.EXE версия 1.00, автор Э. Лантиг

Полноценное антивирусное средство, несмотря на отсутствие некоторых функций приведенной выше идеальной программы. ANTIUS2.EXE отличается простотой обслуживания благодаря системе МЕНЮ, которое позволяет изменять параметры работы. Программа способна блокировать непосредственный доступ к дисководу (в обход системы файлов DOS), операцию форматирования, модификацию программ и файлов типа BAT. Она также проверяет, не пытаются ли выполняемые программы присоединиться к ОС. В случае выявления действий, которые пользователь определил как нелегальные, на дисплее появляется предостерегающее сообщение к вопросу о разрешении или запрете выполнения данной операции. Еще одним достоинством программы является встроенный "дневник работы". Сюда записывается информация о всех запускаемых программах и список модифицированных ими файлов.

Недостатком является большой объем ANTIUS2.EXE (50 Кбайт). Кроме того, ее можно вызвать комбинацией клавиш Alt-4 только тогда, когда COMMAND.COM первого уровня ожидает директив. Попытка вызвать программу в других условиях, например при загруженном Norton Commander, активизирует звуковой сигнал, который не прекращается до тех пор, пока программа не начнет работу - в данном случае до выхода из Norton Commander.

### FLUSHOT2

### FLUSHOT3

### FSP версия 1.00

### FSP версия 1.41, автор Росс М. Гринберг

Очередные версии популярной программы-монитора, которая вначале распространялась как общедоступная (Public Domain), а в последней версии (FluShot Plus) является продуктом типа shareware с низкой регистрационной оплатой (10 дол.). Программа защищает от непосредственного доступа к дисководам и от определенных в специальном конфигурирующем файле таких операций, как:

\* запуск программы, контрольная сумма которой не соответствует сумме, записанной в конфигурирующем файле. Все суммы проверяются также в момент запуска программы FSP;

\* попытка загрузки резидентной программы, которая не содержится в конфигурирующем файле;

\* запись или чтение определенных файлов.

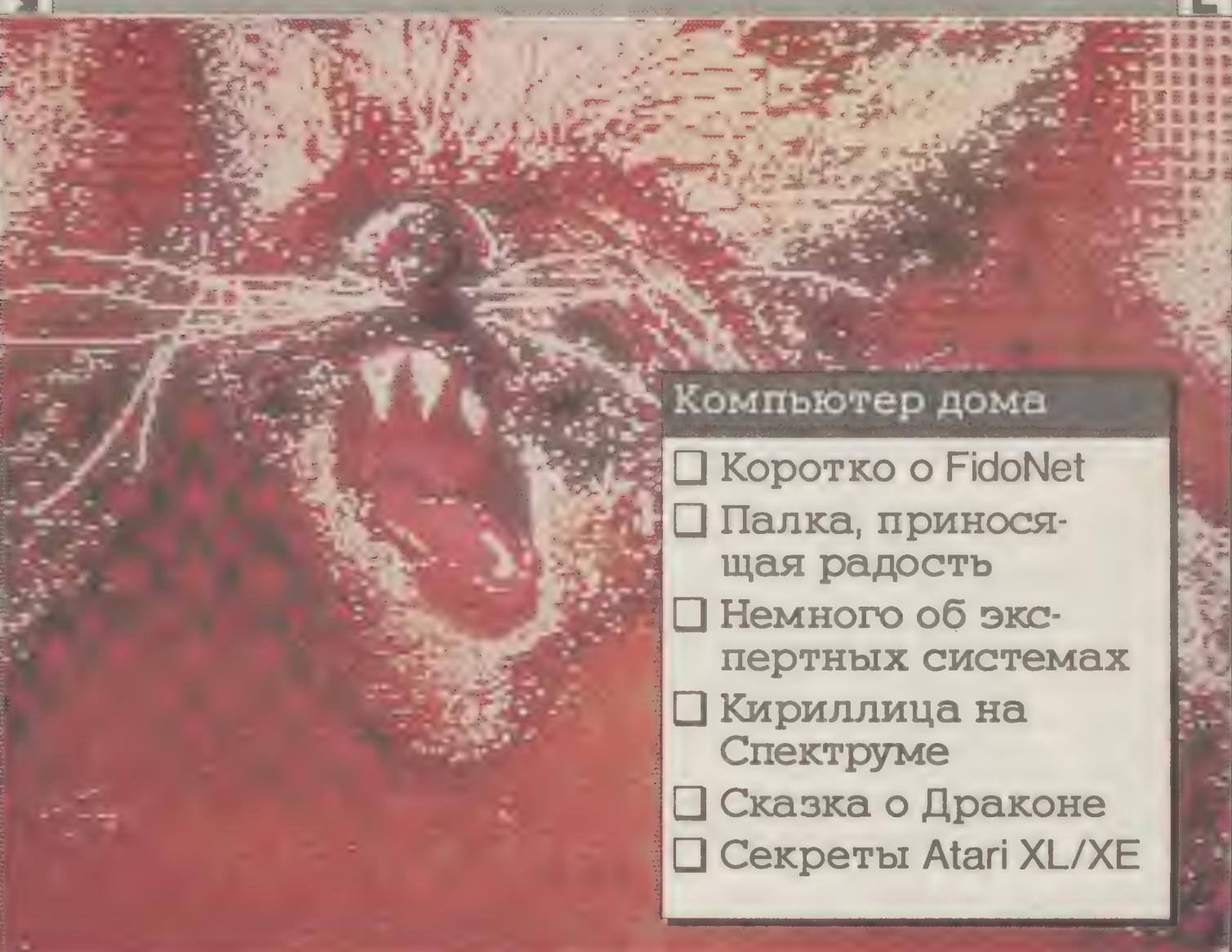
Программа небольшая (10-15 Кбайт) и относительно "умная". К ее недостаткам можно отнести отсутствие контроля за прерыванием BIOS номер 40h, разрешающим запись, а также довольно сложную процедуру предварительного вычисления контрольных сумм. Однако это не уменьшает ценности программы, которая считается одной из лучших в своем классе.

FLUSHOT 2 и 3 отличаются от предыдущих версий отсутствием конфигурирующего файла, в связи с чем список защищаемых программ является постоянным. Кроме того, ими не вычисляются контрольные суммы.

Я представил здесь лишь те программы, которые у меня есть. Безусловно, их гораздо больше. Замечания, сделанные в начале статьи, помогут читателям оценить их качество самостоятельно.

Перевод Анджея Поплавского





## Компьютер дома

- Коротко о FidoNet
- Палка, приносящая радость
- Немного об экспертных системах
- Кириллица на Спектруме
- Сказка о Драконе
- Секреты Atari XL/XE

Компьютер дома

© Ян Стожек

## Коротко о FidoNet

FidoNet является в настоящее время крупнейшей в мире любительской сетью ЭВМ. Эта сеть отличается тем, что для организации так называемого узла не требуется специального оборудования - достаточно иметь компьютер класса PC, модем и две-три простые программы. Несмотря на скромность средств, возможности сети огромны: обмен информацией между узлами обеих Америк, Европы (включая Израиль) и Австралии. Учитывая возможности современной связи, скорость передачи информации не потрясает - сообщение попадает к адресату в течение 1-2 ночей. Но если помнить, что передача корреспонденции ведется только 1 час в сутки (остальное время предназначено для других целей), то результат выглядит не столь уж скромно.

Идея основана на использовании для связи между узлами существующей телефонной сети. В большинстве узлов действуют также BBS (Bulletin Board System) - электронные "почтовые ящики", доступные для всех владельцев компьютеров и модемов. В BBS пользователи могут оставлять свои сообщения, обмениваться программами и т.д. Большая часть суток отводится именно на работу BBS.

Очевидно, что нормальная работа столь огромного организма, как всемирная компьютерная сеть, не была бы возможна без некоторой структуры. Постепенно, по мере расширения, FidoNet также становилась "солидной" организацией. Из простой аморфной сети, в которой каждый мог непосредственно связаться с каждым, она превратилась в иерархическую структуру.

Основным элементом сети является "узел" (NODE). Узлом может быть соответствующим образом оснащенный компьютер, фигурирующий в списке NodeList, который является одним из официальных документов FidoNet и публикуется в каждую пятницу.

С каждым узлом связан SysOp (System Operator) - "оператор узла", т.е. человек, отвечающий за то, чтобы узел работал четко и бесконфликтно (по отношению к другим узлам).



Группа узлов, расположенных в непосредственной близости друг к другу (чаще всего в одном городе или даже районе), составляет "сеть" (NET). Внутри сети обмен данными осуществляется путем непосредственной телефонной связи между узлами, а со всем миром - через выделенный узел, называемый "патроном" (HOST) сети.

Сети, расположенные внутри одного государства (в Европе) или другом крупном регионе, вместе с узлами, не входящими ни в какую сеть (например, по причине удаленности от других узлов), образуют "регион" (REGION). В отличие от сети, регион не посредничает в обмене сообщениями. Он участвует в распространении официальных документов FidoNet, а также занимается вопросами административного характера на своей территории.

Очередным уровнем в иерархии является "зона" (ZONE). Зона приблизительно соответствует континенту. Связь внутри зоны осуществляется по тем же принципам, а обмен между зонами идет через выделенные узлы, называемые "шлюзами" (ZONEGATE). Каждый шлюз обменивается сообщениями только с одной из внешних зон.

Во главе каждого уровня стоит "координатор" (NET, REGION, ZONE COORDINATOR). Он отвечает за то, чтобы "его" структура работала наиболее выгодным для себя и всей FidoNet образом. Координатор сообщает о всех изменениях в структуре координатору вышестоящего уровня (так, чтобы информация могла войти в ближайшее издание списка узлов), а также передает полученную от него информацию координаторам и узлам низших уровней. Координатор занимается также урегулированием конфликтов между операторами узлов и множеством всяких других дел.

FidoNet имеет свой устав, который носит название POLICY. В этом документе определяются основы структуры сети, обязанности и права операторов узлов и координаторов и т.д.

Кроме устава, существуют еще технические нормы - FTSC (Fido Technical Standard Committee), определяющие, в частности, протоколы передачи, формат сообщений, передаваемых через сеть и т.п.

После ознакомления со всей структурой возникает прозаический вопрос - зачем все это нужно? В связи с этим мне вспоминается следующее: на одной ни то конференции, ни то выставке в Москве выступал гость из западной страны. После пространного доклада о компьютерах и модемах кто-то из зала задал ему вопрос: "Ну ладно, только зачем все это?". Докладчик задумался, а потом не очень уверенно ответил: "Ну, например, можно переслать знакомому программу..." Об этом мне рассказал один из редакционных коллег. Я на это ему ответил: "Прекрасно, а вы могли бы объяснить, зачем существует телефон? Или авторучка?". Ответ, казалось бы, совершенно очевиден, но для того, кто не знаком с этими предметами, все не так уж просто.

Точно так же дело обстоит с FidoNet. Это средство, с помощью которого люди разных национальностей, живущие в разных уголках земного шара, могут вести ничем не ограниченный диалог в соответствии с собственными потребностями. В отличие от спутникового телевидения (которое также может принимать каждый) FidoNet позволяет установить дву- и многостороннюю связь без ненужных посредников, модифицирующих информацию в соответствии с собственными представлениями. В отличие от обычной телефонной связи здесь не обязательно, чтобы абонент был дома, когда ему звонят. Кроме того, это более дешевый вид связи: в системе "человек-компьютер-модем" медленнее всего работает человек. С точки зрения пользователя FidoNet больше всего напоми-

нает что-то среднее между телексом и коротковолновой связью, хотя можно найти и другие аналогии. FidoNet - это форма связи, которая не требует от пользователя специализированных знаний: ознакомившись с относительно простыми коммуникационными программами и умев пользоваться телефоном и клавиатурой, можно добиться результатов, о которых до недавнего времени нечего было и мечтать.

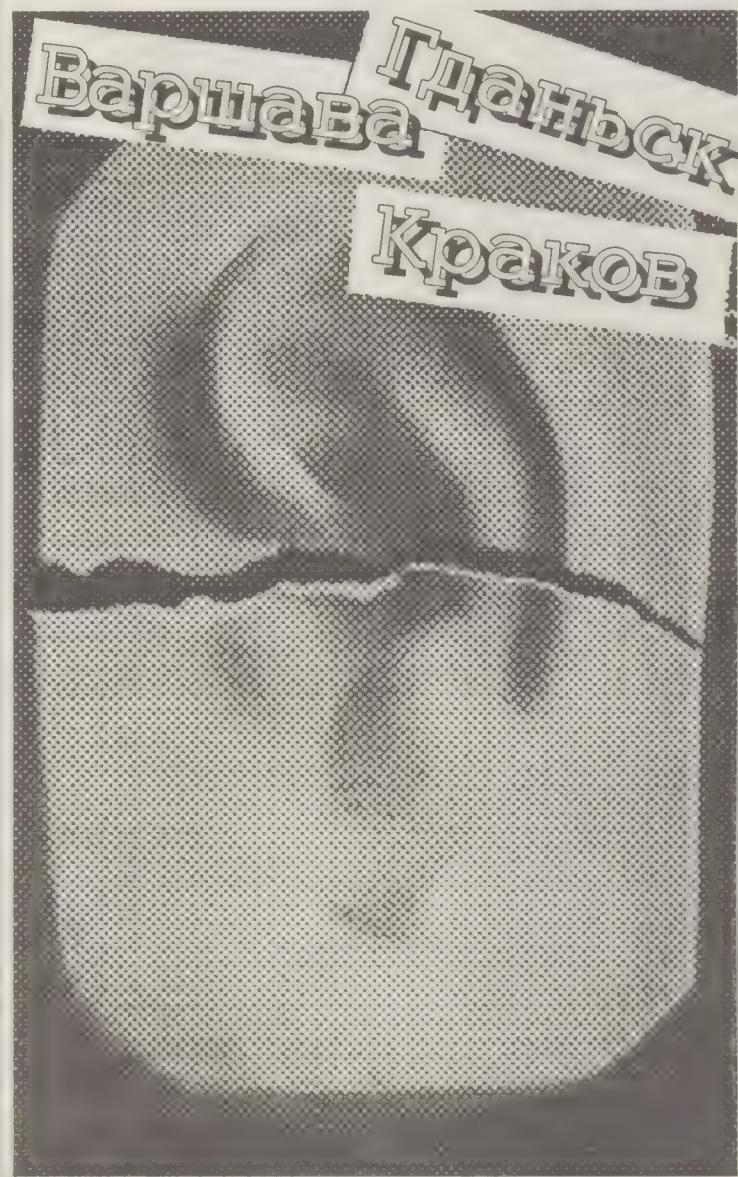


В Польше в настоящее время работают четыре BBS. Один из них (Quick-FIDO BBS) находится в варшавской редакции "Компьютера", второй (QuickCat BBS) - в гданьской фирме "CONTACT", третий (DRILL BBS) - в Кракове и четвертый (MONTH BBS) - также в Варшаве.

Процедура установления связи во всех случаях выглядит так:

- параметры модема: 300 или 1200 бит/сек (в Гданьске можно 2400), без бита четности, 8 бит данных и 1 стоп-бит (т.е. сокращенно 300/1200/2400 N81);
- режим работы модема: ССТТ (команда AT&T);
- набираем один из приведенных ниже номеров... и готово!

BBS представляется, а затем просит сообщить имя и фамилию (можно добавить и отчество, но тогда отделяйте его от фамилии точкой, например, Иван Иванович Иванов. Или просто Иван Иванов - как хотите). При первом сеансе связи выберите пароль, а затем



ответьте на несколько вопросов: хотите ли вы, чтобы BBS демонстрировал вам текст в цвете (с оттенками на монохромном дисплее), очищать ли экран, разбивать ли демонстрируемые тексты на страницы и по сколько строк. Теперь еще осталось заполнить анкету (SysOp хотел бы знать, кто к нему звонит) и можно перейти к главному меню. А здесь перед вами возможность связи со всем миром.

В Польшу можно звонить по следующим телефонам:

Название BBS	Город	Адрес в Fido	Телефон*
QuickCat BBS	Гданьск	2:480/1	52 33 19
QuickFIDO BBS	Варшава	("Компьютер") 2:480/2	21 19 85
DRILL BBS	Краков	2:480/3	21 00 96
MONTH BBS	Варшава	2:480/4	29 15 78

\* Необходимо набрать также номер выхода на международные линии и код города.



Связь с QuickFIDO может выглядеть следующим образом. Когда вы наберете номер, компьютер "поднимет трубку", покажет рисунок собачки вместе с системной информацией и попросит: "Press <Esc> or <B> for BBS...". Если будет нажата нужная клавиша (или спустя примерно 20 секунд), то появится сообщение: "Loading QuickFIDO, please wait...", а после очередной порции информации компьютер запросит имя и фамилию. Если FIDO вас знает, то запросит пароль. Затем появятся местные новости, т.е. сообщения о том, что произошло за последнее время. После того, как вы их прочтете (не обязательно целиком - достаточно нажать клавишу <S>), появится обложка бюллетеня, который содержит сообщения на разные темы. Можно, например, запросить информацию о самых активных пользователях. После просмотра бюллетеня нажмите клавишу <ENTER>, чтобы перейти к главному меню. Теперь опять можно получить немного статистической информации: клавиша <A> (Activity Graph) вызывает график активности, который поможет выбрать самое удобное для выхода на связь время. Затем с помощью клавиши <M> (Mail Menu) можно вызвать "почтовый ящик". Нажав клавишу <L> (Local Letters), а затем <R> (Read Letters) и <F> (Forward) вы сможете прочесть оставленные пользователями сообщения. Они написаны главным образом по-польски. Возвращаемся к "почтовому ящику" (клавиши <S> - Stop Reading и <P> - Previous Menu) и выбираем один из режимов работы EchoMail - например, INTERUSER (свободная переписка между пользователями). Выбор осуществляется клавишами <3>, <R> и <F> - и можно читать. В EchoMail почти все письма написаны по-английски. Это самый распространенный язык в FIDO. Когда вам надоест их читать, достаточно нажать <S> и <P> (главное меню), а затем <F> (File Menu), чтобы попасть в базу файлов. В ответ на <?> FIDO выдаст сообщение о новинках, а после выбора отдела, в котором находится интересующая нас программа, нажатия клавиши <D> (Download) и установления протокола перешлет ее вам. По завершении передачи нажмите <F> (GoodBye) и <Y> (подтверждение). Конец связи. Всего несколько минут, а сколько удовольствия и как полезно!

Перевод Анджея Поплавского

«Компьютер дома»

© Анджей Поплавский

# Палка, приносящая радость

Для всех владельцев 8-разрядных домашних компьютеров джойстик - старый знакомый. Его не столь часто можно увидеть рядом с профессиональными ПК, хотя в последнее время наблюдается своеобразная мода на джойстики для машин класса IBM PC. Давайте же еще раз взглянем на джойстик: может быть, пользуясь им уже долгое время, мы знаем далеко не все об этом чрезвычайно простом и одновременно исключительно удобном устройстве?

Начнем с самого начала - с названия. Если вы заглянете в англо-русский словарь, то найдете следующее: joystick - ручка (рычаг) управления (самолетом). Ну, а если разделить слово на составные части? Получится вот что: joy - радость, stick - палка.. Такая интерпретация не менее близка любителям компьютерных игр, чем "рычаг управления". Вот вам и "палка, приносящая радость". Конструкция джойстика проста, как его название. Посмотрите на принципиальную схему простейшего джойстика (рис. 1). Ее поймет даже тот, кто, завида какую-либо электрическую схему, тотчас переворачивает страницу.

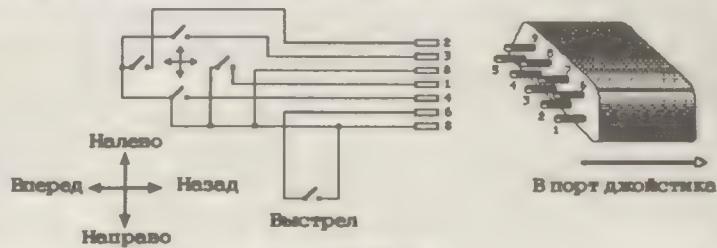


Рис. 1

Комментарии, собственно говоря, излишни. Следует только добавить, что в некоторых компьютерах провода в разъеме, к которому подключается джойстик, расположены иначе, чем в большинстве конструкций. Но в инструкциях об обслуживании компьютеров обычно описываются разъемы, благодаря чему можно буквально за несколько минут "подогнать" джойстик под требования ЭВМ.

Простейший джойстик, основанный на приведенной выше схеме, можно без труда сделать в домашних условиях. Приведу пример своей конструкции - примитивной, но выдержавшей уже два года интенсивной эксплуатации. Отсутствие на рисунке каких-либо размеров связано с тем, что они зависят от того, какие элементы и материалы имеются под рукой.

Я использовал следующие элементы и материалы: кусок толстого текстолита, пять 5-миллиметровых



винтов с гайками, металлический кружок с отверстием по центру, резиновое кольцо, металлическую трубку, микропереключатель (любого типа; важно только, чтобы после снятия пальма он возвращался в исходное положение), рукоять от лыжной палки, 9-контактный штеккер (соответствующий разъему в компьютере, международное обозначение D), пластмассовую коробку для корпуса и соответствующее количество проводов.

Вместо длинного и запутанного описания конструкции предлагаю схемы на рис. 2 и 3. Полагаю, что они достаточно ясны.

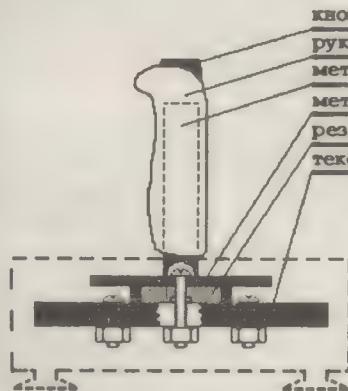


Рис. 2



Рис. 3

Итак, джойстик готов. Если вся электрическая часть выполнена правильно, то он сразу начнет работать. Это, конечно, простейшая конструкция. Вместо контактов в виде головок винтов можно применить, например, микропереключатели, вместо резинового кольца - пружину, вместо... Словом, можно внести любые изменения и усовершенствовать.

Что касается готовых, "фирменных" джойстиков, то их можно условно разделить на два типа: предназначенные для игр, требующих значительной активности играющего, и предназначенные для компьютерной графики. Джойстики первого типа отличаются большой прочностью. Это, например, Micro Blaster или Star Fighter, а также такие известные всем любителям компьютерных игр конструкции, как семейство джойстиков Quickshot (включая модель Turbo). Для работы с программами, требующими большой точности и спокойствия, больше подойдут SpeedKing и Super Professional. Многие джойстики снабжены присосками и не скользят по столу. Заметим, что некоторые конструкции очень удобны для "быстрых" игр, так как имеют до четырех кнопок "Выстрел" (fire) - две на рукоятке (под большим и указательным пальцами) и две на корпусе. К ним принадлежат упомянутый уже Star Fighter, а также, например, VG 125. И еще одна особенность некоторых моделей - "Autofire". В корпусе джойстика монтируется простая схема, имитирующая многократное нажатие кнопки "Выстрел". Иначе говоря, можно "стрелять длинными очередями".

Попадаются и неудачные конструкции. Например, фирменный джойстик Atari Standard: всего одна кнопка на корпусе, отсутствуют присоски, рукоятка короткая и неудобная, а ее передвижение требует значительных усилий.

Следует отметить, что иногда на рынке появляются конструкции, отличающиеся от "нормальных" джойстиков. Встречаются манипуляторы в виде полусферы и без рукоятки, в форме пистолета с небольшой рукояткой наверху, с небольшим шаром вместо рукоятки и т.д. Обычно они недостаточно удобны и проигрывают по сравнению с типичными джойстиками.

Однако вернемся еще раз к функции "Autofire". Один из читателей польского журнала "Komputer", Анджей Шеленг из г. Тшебиня, предложил в свое время небольшую модификацию джойстика из семейства Quickshot. В нормальном варианте обе кнопки на рукоятке выполняют одну и ту же функцию, а после введения изменения одна кнопка "стреляет одиночными", а вторая - "очередями" (имеется в виду работа при включенной функции "Autofire"). На рис. 4б изменения обозначены жирными линиями.

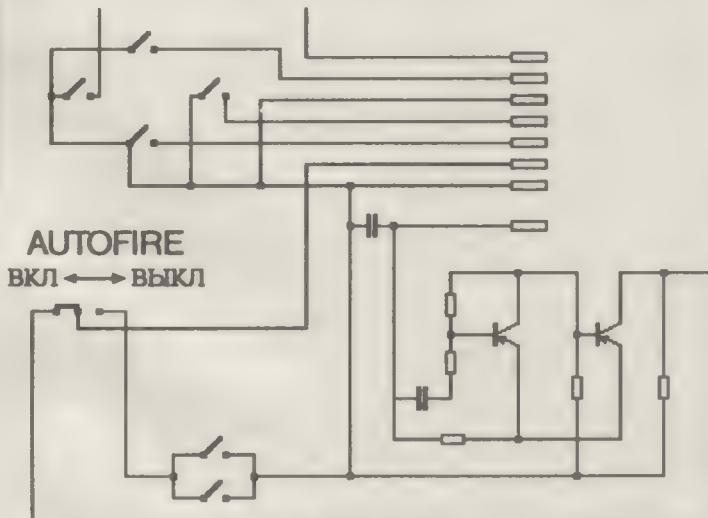


Рис. 4а

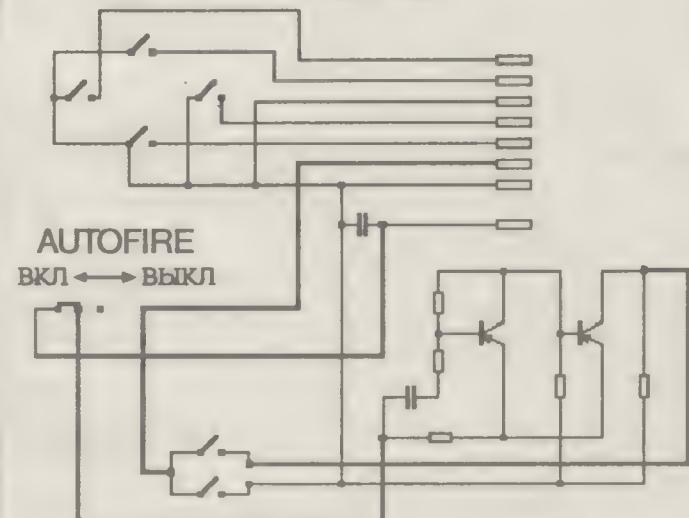


Рис. 4б

Итак, ладонь удобно и уверенно лежит на рукоятке, время старта приближается. Проверьте, свободно ли передвигается рычаг джойстика и - вперед!



# Немного об экспертных системах

Во втором квартале 1990 г. в издательстве "Финансы и статистика" вышла книга Р. Левина, Д. Дранг, Б. Эдлесона "Практическое введение в технологию искусственного интеллекта и экспертных систем". Это книга для начинающих, в которой просто и ясно излагаются ключевые идеи искусственного интеллекта и экспертных систем. Главное достоинство книги в том, что она помогает прейти от идей к их практическому воплощению. Для тех, кто только начал знакомиться с экспертными системами это особенно важно. Получив первые практические результаты, читатель обретет уверенность в себе. В качестве языка реализации экспертных систем в книге выбран Бейсик, наиболее распространенный и простой для освоения язык программирования, интерпретатор или компилятор для которого есть практически на любом компьютере.

Ниже дана сокращениями одна из глав книги.

□ □ □

Существует много способов обучения человека. Можно излагать ему абстрактные факты или учить на примерах. Если сказать ребенку, что дотронувшись до горячей плиты, он сильно обожжется, то вряд ли ребенок сделает из этого вывод, что он обязательно обожжется, дотронувшись до любого горячего предмета. Объяснить ребенку, что нельзя дотрагиваться до горячей плиты можно и другим способом — просто сформулировав общее правило: дотронешься до горячего — обожжешься. Какой бы метод ни использовался при обучении, человек всегда пополняет свои знания, сталкиваясь с чем-то новым. Например, если ребенок еще не понимает, что соприкосновение с горячим предметом приведет к ожогу, и сунул палец в кипяток, то он может наконец прийти к нужному выводу. Такая форма приобретения знаний называется обратной связью. Человек, благодаря механизму обратной связи и уже имеющимся знаниям, приобретает новые. Следовательно, человек учится на опыте.

Приблизительно также "обучается" компьютерная программа. В нее заложен алгоритм хранения фактов и выполнения логических выводов. Логические выводы связаны в программе с постоянно пополняющими ее новыми фактами. Ниже приводится пример самообучающейся системы, который следует рассматривать, исходя из предпосылки:

Система будет самообучаться только в том случае, если она соприкоснулась с чем-то противоречивым ранее известному.

\* Levin R., Drang D., Edelson B. A Comprehensive Guide to AI and Expert Systems. A Tutorial with Many Solved Problems and Programming Examples in the BASIC Language. — McGraw-Hill Book Company, USA, 1986.

По такому же принципу обучается и человек. В конце концов он изменяет свое мнение, если знает или чувствует, что оно ошибочно.

Рассмотрим систему, которая должна понимать различия между автомобилем и танком. Танк и автомобиль обладают, например, такими признаками:

- у танка есть пушка и люк;
- у автомобиля есть дверцы и колеса;
- у танка и у автомобиля есть кузов.

Компьютерную программу можно обучить, объяснив ей разницу только между двумя объектами. Но затем заложенный в нее принцип работы она может распространять на большее число объектов. В программе содержится 3 списка: в списке 1 хранятся атрибуты автомобиля, в списке 2 — танка, а в списке 3 — атрибуты, общие и для танка, и для автомобиля.

Заметим, что система будет работать одинаково независимо от того, о каких объектах пойдет речь. Обучающаяся программа прежде всего запросит у пользователя имена объектов:

ВВЕДИТЕ ИМЯ ОБЪЕКТА 1 ? АВТОМОБИЛЬ  
ВВЕДИТЕ ИМЯ ОБЪЕКТА 2 ? ТАНК

Прежде чем программа приступит к описанию одного из объектов, она запросит ввод его атрибутов:

ВВЕДИТЕ АТРИБУТЫ ЛЮБОГО ОБЪЕКТА И <ВК> В КОНЦЕ

ВВЕДИТЕ АТРИБУТ 1 ? ПУШКА  
ВВЕДИТЕ АТРИБУТ 2 ? ЛЮК  
ВВЕДИТЕ АТРИБУТ 3 ? КУЗОВ  
ВВЕДИТЕ АТРИБУТ 4 ? <ВК>

Затем программа начинает сравнивать введенные атрибуты с атрибутами первого списка, описывающими автомобиль. Поскольку программа еще ничего не знает об автомобиле, первый список пуст и результатом сравнения будет оценка нуль, показывающая, что совпадений нет. По тем же причинам и сравнение с атрибутами второго списка, описывающими танк, даст нулевую оценку. Получив нулевую оценку, программа предположит, что введенные атрибуты относятся к списку 1. Она запросит у пользователя подтверждения этого предположения. Пользователь ответит отрицательно, поскольку это атрибуты не автомобиля, а танка. В системе сработает механизм обратной связи, так как предположение оказалось ошибочным. Пользователь сообщил программе, что введенные атрибуты не относятся к автомобилю и она оставит список 1 пустым. В общий список также ничего не будет занесено, все атрибуты войдут в список 2, где хранятся атрибуты танка. В дальнейшем как только в список 1 будет занесен атрибут, совпадающий с атрибутом списка 2, он будет удален из этих списков и перенесен в общий список. Программа полагает, что совпадающие в списках атрибуты принадлежат общему списку и выполнит все действия автоматически. После первой операции списки будут иметь вид:

Список 1 (автомобиль)	Список 2 (танк)	Список 3 (общий)
ПУШКА		
ЛЮК		
КУЗОВ		

Почему бы не разработать систему так, чтобы она сразу помещала атрибуты в правый список, избегая дополнительных действий? Но именно этим и отличается система, в которую уже заложены все знания, от системы, способной к обучению. В предложенном варианте система ведет себя как человек, а не как машина и должна сама разобраться, куда поместить атрибуты.



Теперь введем атрибуты автомобиля:

ВВЕДИТЕ АТРИБУТЫ ЛЮВОГО ОБЪЕКТА И <ВК> В КОНЦЕ

ВВЕДИТЕ АТРИБУТ 1 ? ДВЕРЦЫ

ВВЕДИТЕ АТРИБУТ 2 ? КОЛЕСА

ВВЕДИТЕ АТРИБУТ 3 ? КУЗОВ

ВВЕДИТЕ АТРИБУТ 4 ? <ВК>

Программа приступит к сравнению введенных атрибутов с атрибутами списка 1 (автомобиль). Результатом сравнения опять будет оценка нуль, поскольку список 1 все еще пуст. Затем при сравнении со списком 2 программа обнаружит совпадение атрибута КУЗОВ. Список атрибутов танка получит оценку 1. Программа предположит, что речь идет о танке и запросит подтверждения у пользователя:

ЭТО ТАНК ? НЕТ

Предположение опять оказалось ошибочным и снова сработает механизм обратной связи. Система увидит, что атрибуты ДВЕРЦЫ, КОЛЕСА и КУЗОВ не относятся к танку, и поместит атрибут КУЗОВ (поскольку он уже есть в списке 2) в общий список (где еще ничего нет). Оставшиеся два атрибута будут занесены в список 1, описывающий автомобиль. После этого списки примут вид:

Список 1 (автомобиль)	Список 2 (танк)	Список 3 (общий)
ДВЕРЦЫ	ПУШКА	КУЗОВ
КОЛЕСА	ЛЮК	

Теперь, приступая к диалогу с пользователем, программа будет делать верные предположения, поскольку, получив от пользователя значения атрибутов и пользуясь механизмом обратной связи, система научилась различать автомобиль и танк. Еще раз кратко опишем основные шаги обучения системы:

1. Система "вносит" изменения только в том случае, если встречаются противоречия. Это обеспечивается с помощью механизма обратной связи.

2. Система сравнивает введенные атрибуты с атрибутами, хранящимися в ней, пытаясь их идентифицировать и выбрать наиболее похожие. Для этого используются специальные оценки. Вся работа выполняется с помощью механизма логического вывода.

Рассмотрим пример программы, написанной на Бейсике, который иллюстрирует основные концепции самообучающихся систем.

Для списков 1, 2 и 3 в программе отведены массивы L1, L2 и LC соответственно. При инициализации в предложениях от 30 до 50 массивам присваиваются пустые значения. Имена объектов вводятся в предложениях 60-70. Атрибуты неизвестного объекта вводятся в предложениях 90-130. Самообучающаяся программа пытается идентифицировать объект по этим атрибутам в предложениях со 180 по 220. Идентификация выполняется с помощью некоторой оценки. Оценка получается из сравнения введенных атрибутов с атрибутами каждого объекта. Предполагается, что введенные пользователем атрибуты принадлежат объекту с наивысшей оценкой (предложение 240). При нулевой оценке выбирается объект 1. Если это предположение верно и в списке атрибутов объекта или общем списке введенных атрибутов еще нет, то они заносятся в список объектов. Атрибуты удаляются из списков объектов и заносятся в общий список в предложениях 330-410 для первого объекта и 780-860 для второго. Остальные операции для первого объекта выполняются в предложениях 420-560, а для второго - в предложениях 920-990.

```

1 REM ** САМООБУЧАЮЩАЯСЯ СИСТЕМА **
2 REM ПРОГРАММА ВЫДВИГАЕТ ПРЕДПОЛОЖЕНИЕ ОВ
ОБЪЕКТЕ
3 REM ИСХОДЯ ИЗ ВВЕДЕННЫХ АТРИБУТОВ
10 REM БЛОК ИНИЦИАЛИЗАЦИИ ДЛЯ
11 REM СПИСКА ОБЪЕКТА 1(L$),
12 REM СПИСКА ОБЪЕКТА 2(O2$) И
13 REM ОВЩЕГО СПИСКА (LC$)
14 REM T$ - ВРЕМЕННЫЙ БУФЕР ДЛЯ ЗНАЧЕНИЙ
АТРИБУТА
20 DIM L$(10),O2$(10),LC$(10),T$(11)
30 FOR H=1 TO 10
40 L$(H)=":":O2$(H)=":":LC$(H)=":"
50 NEXT H
60 INPUT "ВВЕДИТЕ ИМЯ ОБЪЕКТА 1":O1$
70 INPUT "ВВЕДИТЕ ИМЯ ОБЪЕКТА 2":O2$
80 PRINT "ВВЕДИТЕ АТРИБУТЫ ЛЮВОГО ОБЪЕКТА И <ВК> В
КОНЦЕ"
90 FOR H=1 TO 11:T$(H)=":":NEXT H
95 FOR H=1 TO 11
100 PRINT "ВВЕДИТЕ АТРИБУТ":H
110 INPUT T$(H)
120 IF T$(H)=":": THEN GOTO 140
130 NEXT H
140 REM ЧИСЛО АТРИБУТОВ МЕНЬШЕ НА 1, ТАК КАК ВЫЛ
ВКЛЮЧЕН 0
150 L=1
160 REM СРАВНЕНИЕ АТРИБУТОВ СО СПИСКАМИ
ОБЪЕКТОВ 1 И 2
170 REM СНАЧАЛА УСТАНОВИТЬ ОЦЕНКУ ДЛЯ ОБЪЕКТОВ 1
И 2 В 0
180 S1=0:S2=0
190 FOR H=1 TO L
195 FOR K=1 TO 10
200 IF O1$(H)=T$(K) THEN S1=S1+1
210 IF O2$(H)=T$(K) THEN S2=S2+1
215 NEXT K
220 NEXT H
230 REM ОБЪЕКТ 2 ПРЕДПОЛАГАЕТСЯ ТОЛЬКО, ЕСЛИ
ОЦЕНКА 2 ВОЛЬШЕ
231 REM ОЦЕНКИ 1
240 IF S1 < S2 THEN GOTO 700
250 PRINT "ЭТО":O1$
260 INPUT "ВЕРНО ? ДА ИЛИ НЕТ":S
270 IF S=":": THEN GOTO 700
280 REM ПОМЕСТИТЬ КАЖДЫЙ АТРИБУТ ВО ВРЕМЕН
НЫЙ БУФЕР,
290 REM А НЕ В СПИСОК ОБЪЕКТОВ 1 ИЛИ 2 ИЛИ ОВЩИЙ
СПИСОК
310 REM ПЕРЕМЕСТИТЬ АТРИБУТ ИЗ СПИСКА 2 В ОВЩИЙ
СПИСОК,
320 REM ЕСЛИ ЭТО НЕОВХОДИМО
330 FOR H=1 TO L
340 FOR K=1 TO 10
350 IF T$(H)=O2$(K) THEN GOTO 400
360 REM ПЕРЕМЕСТИТЬ АТРИБУТ ИЗ ВТОРОГО СПИСКА В
ОВЩИЙ
370 FOR L=1 TO 10
380 IF LC$(L)=":": THEN GOTO 390
382 LC$(L)=O2$(K):O2$(K)=":":T$(H)=":":390 NEXT L
400 NEXT K
410 NEXT H
420 FOR H=1 TO L:REM УКАЗАТЕЛЬ НА ВРЕМЕННЫЙ БУФЕР
430 FOR K=1 TO 10:REM УКАЗАТЕЛИ НА ОВЩИЙ СПИСОК И
СПИСОК

```

```

430 REM ОВЪЕКТА 1
440 IF T$(I)=LC$(K) THEN GOTO 520: REM ЕСТЬ В ОВЩЕМ
СПИСКЕ
450 IF T$(I)=O1$(K)=":T$(I)=": REM ЕСТЬ В СПИСКЕ ОВЪЕКТА 1
460 NEXT K
470 REM ПОМЕСТИТЬ ПЕРЕМЕННУЮ В СПИСОК 1
480 REM ПОИСК СВОВОДНОГО МЕСТА ПОД ПЕРЕМЕННУЮ
490 FOR K=1 TO 10
495 IF T$(I)="" THEN GOTO 510
500 IF O1$(K)="" THEN O1$(K)=T$(I):GOTO 520
510 NEXT K
520 NEXT I: REM СЛЕДУЮЩИЙ АТРИБУТ
530 REM ПЕЧАТЬ ВСЕХ СПИСКОВ
540 GOSUB 2000
550 REM СЛЕДУЮЩИЕ АТРИБУТЫ
560 GOTO 80
700 PRINT "ЭТО ";O2$
710 INPUT "ВЕРНО ДА ИЛИ НЕТ";I$
720 IF I$="НЕТ" THEN GOTO 330
740 REM ПОМЕСТИТЬ КАЖДЫЙ АТРИБУТ ВРЕМЕННОГО БУ-
ФЕРА, КОТОРОГО
750 REM НЕТ В ОВЩЕМ СПИСКЕ И В СПИСКЕ ОВЪЕКТА 2, В
СПИСОК 2
760 REM ПЕРЕМЕСТИТЬ СОВПАДАЮЩИЕ АТРИБУТЫ ИЗ
СПИСКА 1
770 REM В ОВЩИЙ СПИСОК
780 FOR I=1 TO L
790 FOR K=1 TO 10
800 IF T$(I)=O1$(K) THEN GOTO 850
810 REM ПЕРЕМЕСТИТЬ АТРИБУТ ИЗ СПИСКА 1 В ОВЩИЙ
СПИСОК
820 FOR L=1 TO 10
830 IF LC$(L)="" THEN GOTO 840
832 LC$(L)=O1$(K):O1$(K)=":T$(I)="
840 NEXT L
850 NEXT K
860 NEXT I
870 FOR H=1 TO L: REM УКАЗАТЕЛЬ НА ВРЕМЕННЫЙ БУФЕР
880 FOR K=1 TO 10: REM УКАЗАТЕЛЬ НА ОВЩИЙ СПИСОК И
СПИСОК 2
890 IF T$(I)=LC$(K) THEN GOTO 970:REM ЕСТЬ В ОВЩЕМ
СПИСКЕ
900 IF T$(I)=O2$(K) THEN GOTO 970:REM ЕСТЬ В СПИСКЕ 2
910 NEXT K
920 REM ПОМЕСТИТЬ ПЕРЕМЕННУЮ В СПИСОК 2
930 REM ПОИСК СВОВОДНОГО МЕСТА ДЛЯ ПЕРЕМЕННОЙ
940 FOR K=1 TO 10
945 IF T$(I)="" THEN GOTO 960
950 IF O2$(K)="" THEN O2$(K)=T$(I):GOTO 970:960 NEXT K
970 NEXT I:REM ОВЩИЙ АТРИБУТ
980 REM ОВЩАЯ ПЕЧАТЬ
990 GOSUB 2000
1000 REM СЛЕДУЮЩИЙ АТРИБУТ
1010 GOTO 80
2000 REM ПЕЧАТЬ ВСЕХ СПИСКОВ
2010 PRINT O1$,O2$,"ОВЩИЙ СПИСОК"
2020 FOR I=1 TO 10
2030 PRINT O1$(I),O2$(I),LC$(I)
2040 NEXT I
2050 RETURN

```

Перевод с английского Михаила Сальникова, Юлии Сальниковой.

«Компьютер дома»

© Владислав Крамаренко

# Кириллица на Спектруме

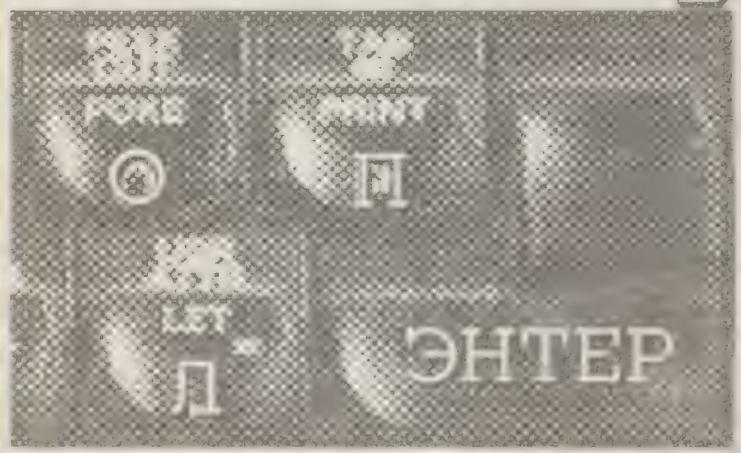
Предлагаемая вашему вниманию программа предназначена для микрокомпьютеров, совместимых по программному обеспечению с домашним компьютером ZX Spectrum. Коды знаков русских букв соответствуют стандарту КОИ-7. Выбранная форма знаков обеспечивает хорошую разборчивость текста и низкую утомляемость даже при работе на дисплеях, параметры которых оставляют желать лучшего.

Коды знаков занимают 768 байт и могут располагаться в свободной части оперативной памяти. Если RAMTOP установлен ниже начального адреса массива кодов алфавита, то команда NEW, стирающая BASIC-программы, не затронет массива кодов. RAMTOP устанавливается с помощью команды CLEAR xxx, где xxx равно начальному адресу массива кодов минус единица.

В программу входят загрузчик (строки 5-60), преобразующий шестнадцатеричные коды в десятичные, и массив данных. Каждая строка массива состоит из данных одного знака и команды-рекомандации (REM), за которой следует буква латинского алфавита, превращающаяся после запуска программы в букву русского алфавита. По начертанию этой буквы можно судить о правильности набора строки.

Наберите программу и запустите ее командой RUN. С помощью команды LIST проверьте начертание букв. Если обнаружите ошибки, исправьте их и снова запустите программу. Сохранить массив кодов на магнитную ленту вы можете с помощью команды SAVE "R.ALF" CODE 64000,768.

Считывание с ленты осуществляется командой LOAD "R.ALF" CODE либо LOAD "R.ALF" CODE n, где n - новый адрес, который должен быть кратен числу 256. Переключение с русского алфавита на латинский осуществляется командой POKE 23607,60, а с латинского на русский - POKE 23607,x, где x - адрес, разделенный на 256, минус 1.



1 REM R.ALPHABET  
 3 CLEAR 63999  
 5 LET A=10: LET B=11: LET C=12  
 10 LET d=13: LET e=14: LET f=15  
 12 LET x=64000: POKE 23607,249  
 15 READ a\$  
 20 FOR n=1 TO 15 STEP 2  
 30 LET w=VAL a\$(n)  
 40 LET w=w\*16: LET w=w+VAL a\$(n+1)  
 50 POKE x,w: LET x=x+1  
 60 NEXT n: GO TO 15  
 9903 DATA "0000000000000000": REM -  
 9904 DATA "1818181818001800": REM !  
 9905 DATA "6C6C000000000000": REM -  
 9906 DATA "6C6CFE6CFE6C6C00": REM #  
 9907 DATA "187EE07C0EFC3800": REM \$  
 9908 DATA "00C6CC183066C600": REM %  
 9909 DATA "386C3877DECE7B00": REM &  
 9910 DATA "1818300000000000": REM '  
 9911 DATA "000C1818180C00": REM (   
 9912 DATA "0030181818183000": REM )  
 9913 DATA "00663CFF3C660000": REM \*  
 9914 DATA "0018187E18180000": REM +  
 9915 DATA "0000000018183000": REM ,  
 9916 DATA "00000007E00000000": REM -  
 9917 DATA "0000000000181800": REM .  
 9918 DATA "03060C183060C000": REM /  
 9919 DATA "7CC6CED6E6C67C00": REM Ø  
 9920 DATA "18381818183C00": REM 1  
 9921 DATA "3C66061C30667E00": REM 2  
 9922 DATA "3C66061C06663C00": REM 3  
 9923 DATA "0E1E36667F060F00": REM 4  
 9924 DATA "7E607C0606663C00": REM 5  
 9925 DATA "1C30607C66663C00": REM 6  
 9926 DATA "7E66060C18181800": REM 7  
 9927 DATA "3C66663C66663C00": REM 8  
 9928 DATA "3C66663E060C3800": REM 9  
 9929 DATA "0018180018180000": REM :  
 9930 DATA "0018180018183000": REM ;  
 9931 DATA "0C18306030180C00": REM <  
 9932 DATA "00007E00007E0000": REM =  
 9933 DATA "6030180C18306000": REM >  
 9934 DATA "3C66060C18001800": REM ?  
 9935 DATA "0000CEDBFBDCE00": REM @  
 9936 DATA "00003C063E663F00": REM A  
 9937 DATA "043C607C66663C00": REM B  
 9938 DATA "000066666667F03": REM C  
 9939 DATA "00007E666666FFC3": REM D  
 9940 DATA "00003C667E603E00": REM E  
 9941 DATA "00187ED8DBDB7E18": REM F  
 9942 DATA "00007C6060606000": REM G  
 9943 DATA "0000C66C386CC600": REM H  
 9944 DATA "000066666E766600": REM I  
 9945 DATA "180066666E766600": REM J

9946 DATA "00000666C786666600": REM K  
9947 DATA "000000E1E3666C600": REM L  
9948 DATA "00000C6EED6C6C600": REM M  
9949 DATA "0000066667E6666600": REM N  
9950 DATA "000003C66666663C00": REM O  
9951 DATA "000007E6666666600": REM P  
9952 DATA "000003E663E366600": REM Q  
9953 DATA "000007C667C606000": REM R  
9954 DATA "000003C6660663C00": REM S  
9955 DATA "000007E18181800": REM T  
9956 DATA "000006666663E063C": REM U  
9957 DATA "00000DB7E187EDB00": REM V  
9958 DATA "000007C667C637E00": REM W  
9959 DATA "0000060607C667C00": REM X  
9960 DATA "00000C6C6F6DAF600": REM Y  
9961 DATA "000003C660C663C00": REM Z  
9962 DATA "00000D6D6D6D6FE00": REM [  
9963 DATA "000007C063E067C00": REM \  
9964 DATA "00000D6D6D6D6FF03": REM ]  
9965 DATA "0000066663E060600": REM ↑  
9966 DATA "0000000000000000FF": REM \_  
9967 DATA "CEDBD8FBDBDBCE00": REM E  
9968 DATA "0E1E3666FEC6C600": REM a  
9969 DATA "FCC0C0FCC6C6FC00": REM b  
9970 DATA "CCCCCCCCCCCCFC06": REM c  
9971 DATA "7E666666666666FFC3": REM d  
9972 DATA "FCC0C0F8C0C0FE00": REM e  
9973 DATA "187EDBDBDB7E1800": REM f  
9974 DATA "7C6060606060606000": REM g  
9975 DATA "C3663C183C66C300": REM h  
9976 DATA "C6C6CEDEF6E6C600": REM i  
9977 DATA "D6C6CEDEF6E6C600": REM j  
9978 DATA "C6CCD8F8CCC6C600": REM k  
9979 DATA "0E1E3666C6C6C600": REM l  
9980 DATA "C6EEFED6C6C6C600": REM m  
9981 DATA "C6C6C6FEC6C6C600": REM n  
9982 DATA "7CC6C6C6C6C67C00": REM o  
9983 DATA "FEC6C6C6C6C6C600": REM p  
9984 DATA "7EC6C67E3666C600": REM q  
9985 DATA "FCC6C6FCC0C0C000": REM r  
9986 DATA "7CC6C0C0C0C67C00": REM s  
9987 DATA "FC30303030303000": REM t  
9988 DATA "C6C6C67E06C67C00": REM u  
9989 DATA "DBDB7E187EDBDB00": REM v  
9990 DATA "F8CCCCFCC6C6FC00": REM w  
9991 DATA "C0C0C0FCC6C6FC00": REM x  
9992 DATA "C6C6C6F6DADAF600": REM y  
9993 DATA "3C66061C06663C00": REM z  
9994 DATA "D6D6D6D6D6D6FE00": REM {  
9995 DATA "7CC6061E06C67C00": REM i  
9996 DATA "D6D6D6D6D6D6FF03": REM }  
9997 DATA "666663E0606060600": REM ~  
9998 DATA "3C4299A1A199423C": REM ©  
9999 REM START 64000, END 64768

# Сказка о Драконе

Программирование - трудное искусство. Однако самое скверное, что когда программа уже написана, часто оказывается, что она не работает. Надо еще вложить много труда в отладку, но и после нее программа не всегда работает так, как было задумано. Обычно мы добиваемся полного успеха только после продолжительных усилий.

К сожалению, радость от успеха зачастую омрачают комментарии друзей, которые посмеиваются над программой (как будто они сами написали бы ее лучше!), дают ей всякие обидные прозвища и с видом знатоков заявляют, что программа не структурная! Для того чтобы вынести окончательный, уничтожительный приговор, им, оказывается, достаточно только того, что программа написана на Бейсике...

Можно ли избежать этих неприятностей? Конечно, можно. Правда, существует мнение, что Бейсик не подходит для структурного программирования. Может быть это и так, но ведь строение программы зависит прежде всего от программиста. Никому не запрещается, даже на Бейсике, избегать перехода в середину цикла. Можно также каждый раз выполнять подпрограмму заново, не разбивая ее на отдельные фрагменты, можно отказаться начинать выполнение программы с произвольно выбранной внутренней строки. Наконец, пока еще никто никого не заставляет использовать оператор **GOTO** в каждой второй строке.

Если кто-нибудь сомневается, что можно составить совсем неплохую (во всяком случае по мнению автора) программу на Бейсике, которая будет хотя бы частично структурной, пусть вместе с нами введет в свой Spectrum программу-сказку.

Я предлагаю внимательно проследить весь процесс построения программы. Создавая ее, постараемся, чтобы она выглядела как следует.

... Давным-давно жил да был большущий змей. Звали его Драконом, причем не простым, а Драконом-Цифроедом. Его так прозвали потому, что достаточно ему было основательно проголодаться, как выползал он в пустыню и скирал все цифры, которые попадались ему навстречу. Ни одной не пропускал! Они шли ему на пользу: только проглотит Цифроед цифру - тут же поправляется, длиннее становится. Одного он только боялся: что когда-нибудь по ошибке или с голоду сам себя слопает. А это ощущение не из приятных.

Вот такую сказку мы и "загрузим" в наш компьютер, а будет она рассказана на Бейсике и весьма структурно!

Начнем с того, что разобьем задачу на составные части, которые легко будет написать и потом объединить в единое целое.

В игре, которую мы только что задумали, нам придется общаться с компьютером с помощью клавиатуры. Поэтому один кусок программы должен опознавать нажатую нами клавишу, второй - строить на дисплее изображение (при этом предварительно должно быть ликвидировано прежнее - Дракон ведь будет передвигаться!). Нам понадобится подпрограмма, реализующая движение Цифроеда, и еще одна, проверяющая, что с ним произошло. В одной из час-

тей программы должно происходить генерирование новой цифры нельзя же допустить, чтобы Дракон постоянно ходил голодный. Напоследок я оставил то, что должно быть в самом начале: описание используемых в программе переменных и присвоение им начальных значений.

Если бы все только что перечисленные элементы были уже готовы, то нам осталось бы объединить их в единое целое:

```

1 REM Добрый день
10 REM Описание переменных и присвоение им
 начальных значений
99 REM Конец описания
100 REM Начало цикла (заканчивающегося в 160
 строке)
110 GOSUB 1000: REM картинка с Драконом
120 GOSUB 2000: REM Новая цифра
130 GOSUB 3000: REM Общение - управление
140 GOSUB 4000: REM Движение Дракона
150 GOSUB 5000: REM Проверка и реакция
160 IF ESHCHO<>0 THEN GOTO 100: REM условие цикла
170 REM Окончание
180 PRINT "GAME OVER"
190 STOP

```

(Если ваш компьютер все еще сопротивляется и не хочет писать по-русски - не заставляйте его силой, а загляните на 57 страницу. Тогда уж он обязательно захочет - никуда не денется...)

Эта программа - самая что ни на есть структурная. Она содержит описания, один цикл типа **UNTIL** (условие проверяется после каждого выполнения цикла, а цикл повторяется до тех пор, пока условие не будет выполнено) и один оператор после завершения цикла. Здесь нет никаких переходов и неопрятно перемешанных подпрограмм (этого, правда, еще не видно, но, уверяю вас, их не будет).

Имена всех использованных переменных и их начальные значения мы узнаем лишь после того, как составим всю программу. Именно поэтому строки с двузначными номерами введем в конце.

А сейчас давайте займемся самым сложным: рисунком Дракона-Цифроеда, не упуская из виду, что в каждой подпрограмме надо определять локальные переменные.

Я прекрасно знаю, что в Бейсике нет необходимости описывать простые переменные. В некоторых вариантах этого языка (например, в Бейсике ZX Spectrum) описать простую числовую переменную даже невозможно. Однако я считаю, что для наших целей достаточно всего один раз упомянуть переменную в области программы, предназначенней для описаний, а не там, где она используется. Мы ведь вполне в начале программы можем подставить вместо числовых переменных нули, а вместо строковых переменных - пустые последовательности. В результате с самого начала будет известно, какие именно переменные мы собираемся использовать в программе (ведь каждую из них можно снабдить комментарием). Кроме того, если мы будем использовать переменную только для хранения целых чисел, то присвоим ей значение 0, а если и для действительных чисел, то 0.0. Это заменит нам правильное описание переменных.

В языке Бейсик нет локальных переменных, которые были бы доступны в подпрограммах. Однако, по-



скольку нам удалось так пронумеровать строки, что подпрограммы располагаются в разных "тысячах", то давайте воспользуемся этим и всем переменным, используемым только в "х-тысячной" подпрограмме, дадим имена, заканчивающиеся числом *x*. Вот и все, что я хотел сказать об описаниях.

Итак, приступим наконец к рисованию. Допустим, что головой Дракона будет символ "\*", туловище будет состоять из символов "Ф", а хвост - из "х".

Для того чтобы создать впечатление движения на экране, мы каждый раз будем менять положение головы и хвоста Дракона. Предыдущее положение головы заменим туловищем, а след от хвоста сотрем. Необходимо, однако, помнить, что хвост должен "ходить" по следам головы. Для реализации такого движения воспользуемся массивом с именем "W", в котором будут храниться координаты головы после очередного перемещения. Новое положение хвоста мы извлечем из этого же массива. Разница между индексами для головы и хвоста будет равна длине туловища Дракона в данный момент. Эта длина будет меняться после каждой "съеденной" цифры.

Итак, массив "W" должен быть двумерным. Первое измерение равно двум: оно соответствует горизонтальной и вертикальной координатам. Второе измерение будет зависеть от способностей игроков. Пусть для начала оно составляет 100. По ходу игры массив будем заколызовывать. Иными словами, когда нам понадобится записать данные в строку 101, будем записывать их вместо этого в строку с номером 1. Перед тем как их туда записать, нам каждый раз придется проверять, не превысили ли индексы головы и хвоста значение 100? Эту задачу, конечно, будет решать подпрограмма 4000 - движение Дракона. В настоящий момент нам достаточно знать, что глобальные переменные GOLOVA и HVOST определяют текущее положение головы и хвоста в массиве W. Итак, запишем:

```

1000 REM подпрограмма изображения дракона
1010 LET P1=0: REM локальная вспомогательная
переменная
1020 LET P1=GOLOVA-1
1030 IF P1<1 THEN LET P1=100
1040 PRINT AT W(GOLOVA,1),W(GOLOVA,2); "*"
1050 PRINT AT W(P1,1),W(P1,2); "Ф"
1060 IF PRIROST>1 THEN LET PRIROST=PRIROST-1: RE-
TURN REM дракон растет
1070 LET P1=HVOST-1
1080 IF P1<1 THEN LET P1=100
1090 PRINT AT W(P1,1),W(P1,2); " "
1100 PRINT AT W(HVOST,1),W(HVOST,2); "х"
1110 LET PRIROST=0
1120 RETURN

```

Обратите внимание на строку 1060. В ней после THEN использовалась последовательность из двух операторов, разделенных друг от друга двоеточием. Такое решение выгодно в тех случаях, когда нам нужно выполнить всего несколько операторов. Вторым оператором является команда выхода из подпрограммы. Конечно, вполне возможен переход к оператору RETURN в строке 110, но это запутало бы структуру подпрограммы.

Второе замечание касается проверяемого в строке 1060 условия. Проверка на равенство нулю является более естественной, чем проверка неравенства. Однако именно проверка на равенство значительно усложнила бы программу. Хвост рисуется слишком долго и соответствующие команды не умес-

тились бы после THEN (тем более, что среди них есть условный оператор), и нам пришлось бы выполнять какой-нибудь переход (а это, как видите, совершенно не нужно).

Следующая подпрограмма - 2000 - генерирует цифру и выводит ее на экран. Нам потребуются здесь две локальные переменные, в которых мы запомним координаты цифры: ведь нужно быть уверенным, что цифра появится на свободном поле. Значение цифры прочитаем по ее коду ASCII. Для того чтобы генерировать правильные координаты, снова воспользуемся механизмом цикла UNTL.

```

2000 REM подпрограмма, генерирующая цифру
2010 LET P2=0: LET R2=0: REM две локальные пере-
менные из целых чисел
2020 IF ZDES=1 THEN RETURN: REM здесь еще ста-
рая цифра
2030 REM начало цикла определения координат
2040 LET P2=INT(RND*22)
2050 LET R2=INT(RND*32)
2060 IF SCREEN$(P2,R2) = " " THEN GOTO 2040: REM
условие цикла
2070 PRINT AT P2,R2:CHR$(INT(RND*10)+49)
2080 LET ZDES=1
2090 RETURN

```

Теперь примемся за подпрограмму диалога человека с компьютером. Этот диалог необходим нам только (!) для того, чтобы управлять действиями Дракона (я забыл вас предупредить, что после длительного общения с компьютером человек сам превращается в Дракона-Цифроеда и начинает питаться только цифрами).

Давайте выберем для управления движениями Дракона по экрану любые понравившиеся вам клавиши (например, *Q* и *A* - вверх-вниз, а *O* и *P* - влево-вправо) и воспользуемся стандартными процедурами чтения клавиатуры, распознавания и произношения новых значений двум параметрам изменения направлений.

```

3000 REM подпрограмма диалога
3010 LET K$="": REM символ с клавиатуры
3020 LET K$=INKEY$
3030 IF K$="A" THEN LET ZKPI=1: LET ZKPO=0
3040 IF K$="Q" THEN LET ZKPI=-1: LET ZKPO=0
3050 IF K$="O" THEN LET ZKPI=0: LET ZKPO=-1
3060 IF K$="P" THEN LET ZKPI=0: LET ZKPO=1
3070 RETURN

```

После чтения клавиатуры мы можем изменить положение Дракона, т.е. записать в массив W новые значения и передвинуть указатели GOLOVA и HVOST.

```

4000 REM подпрограмма передвижения Дракона
4010 LET P4=0: LET R4=0: REM вспомогательные
переменные
4020 LET P4=P1+ZKPI
4030 LET R4=PO+ZKPO
4040 IF P4>0 AND P4<31 THEN LET P1=P4: REM новое
вертикальное положение
4050 IF R4>0 AND R4<31 THEN LET PO=R4: REM новое
горизонтальное положение
4060 LET GOLOVA=GOLOVA+1: REM перемещение го-
ловы

```



4070 IF GOLOVA>100 THEN LET GOLOVA=1: REM и массив закольцовывается

4080 IF PRIROST=0 THEN LET HVOST=HVOST+1: REM обратите внимание на прирост туловища

4090 IF HVOST>100 THEN LET HVOST=1

4100 LET W(GOLOVA,1)=PI

4110 LET W(GOLOVA,2)=PO

4120 RETURN

Придется еще проверить, съел ли Дракон в данный момент цифру или еще не успел. Для этого применим информацию, заключенную в "экране". Прежде чем нарисовать там Дракона, прочтем код символа, изображенного в соответствующей позиции. Если это будет туловище Дракона, игра заканчивается.

5000 REM подпрограмма, проверяющая эффект перемещения

5010 LET P5=0: REM вспомогательная переменная - код символа на экране

5020 LET P5-CODE(SCREEN\$(PI,PO))

5030 IF P5>48 AND P5<58 THEN LET PRIROST=P5-48: REM см. ниже

5031 REM если здесь была цифра - туловище становится длиннее

5040 IF P5=35 OR P5=42 THEN LET ESHCHO=0: REM см. ниже

5041 REM Дракон укусил сам себя - игра окончена

5050 RETURN

Это, собственно, конец программы. Остается только ввести строки описаний. Это важно, поскольку в противном случае в начале игры некоторые переменные не имели бы значений.

Поскольку вы начали с того, что нарисовали Дракона, надо запомнить в массиве W его исходное положение - разумеется, после определения массива.

10 DIM W(100,2): REM следы дракона

20 LET W(1,1)=10: LET W(1,2)=14

30 LET W(2,1)=10: LET W(2,2)=15

40 LET W(3,1)=10: LET W(3,2)=16: REM здесь появляется дракон

50 LET PI=10: LET PO=16: LET ZKPI=0: LET ZKPO=1: REM см. ниже

51 REM исходное положение и направление движения

60 LET GOLOVA=3: LET HVOST=1

Теперь остается лишь установить исходные значения параметров:

70 LET PRIROST=0: REM прирост туловища дракона

80 LET ESHCHO=1: REM игра еще продолжается

90 LET ZDES=0: REM здесь еще нет цифры

- и игра закончена.

Желаю всем приятно провести время за игрой. И не бойтесь "знатоков"!

Перевод Тадеуша Радюша



ИЗДАТЕЛЬСТВО "ФИНАНСЫ И СТАТИСТИКА" СОВМЕСТНО С РЕДАКЦИЕЙ  
ПОЛЬСКОГО ЖУРНАЛА **Компьютер**  
осуществляет выпуск нового периодического сборника **Компьютер**  
на русском языке.

Издательство "Финансы и статистика" принимает заказы на сборник и  
производит отправку наложенным платежом.

ЗАКАЗ высылается по адресу: 101000 г. Москва, ул. Чернышевского, 7.  
Тел. 925-35-02, 923-18-68.

## ЗАКАЗ

От кого.....

Адрес..... Телефон.....  
(почтовый индекс указывать обязательно)

№ выпусков..... Количество экземпляров журнала "Компьютер".....



POLSIN SYSTEMS PTE LTD  
AL. 3 MAJA 9, 30-062 KRAKOW, ПОЛЬША  
TEL (12) 331100 FAX (12) 331100 TLX 325433

Сингапурско - польская компания

POLSIN SYSTEMS PTE LTD

Предлагает:

- профессиональные компьютеры класса PC/XT, AT, 386
- принтеры любого типа
- различные периферийные устройства
- системное и прикладное программное обеспечение, консультацию в области компьютеризации
- системы приемников спутникового ТВ
- другое электронное оборудование по заказу покупателя.

На дальневосточном рынке мы действуем уже 15 лет - пользуйтесь нашим опытом.

Предлагаем самые низкие цены и проверенных поставщиков, даем гарантию.

Мы готовы к каждому виду сотрудничества!

Наилучшая дорога к рынкам Сингапура, Тайвания, Японии, Южной Кореи, Малайзии, Гонконга - через POLSIN SYSTEMS PTE LTD

# Секреты Atari XL/XE

Каждый счастливый владелец компьютера практически при первой же встрече с языком Бейсик сталкивается с загадочными операторами PEEK и POKE. Попытки применять их наугад иногда дают интересные результаты, иногда не дают ничего, а иногда приводят к сбою системы. Дело в том, что для правильно использования этих операторов необходимо знать функции ячеек памяти и величины, которые можно в них вводить. Напомним, что:

\* POKE (анг. запихивать, заталкивать) – позволяет поместить в определенную ячейку памяти любую величину от 0 до 255;

\* PEEK (анг. заглядывать, подсматривать) – позволяет проверить, какая величина находится в данный момент в определенной ячейке памяти компьютера.

Искусное "жонглирование" этими операторами – дело не простое, но даже начинающий может получить удовлетворение, когда ему удастся выйти за рамки элементарных основ Бейсика. С их помощью можно модифицировать экран (Display List), писать машинные подпрограммы, звуковое сопровождение и т.д.

Здесь мы займемся только адресами и функциями простейших ячеек памяти, которыми может легко воспользоваться каждый. Итак, начинаем:

16 – прерывания POKEY, например POKE 16,255 отключает клавиатуру, POKE 16,64:POKE 53774,64 блокирует клавишу BREAK.

18, 19, 20 – таймер; содержимое ячейки 20 увеличивается на единицу каждую 1/50 секунды (система PAL). При величине 255 наступает обнуление ячейки 20 с одновременным увеличением на единицу содержимого ячейки 19, при величине 255 в ячейке 19 то же самое происходит с ячейкой 18. Ниже дан пример использования в программе подсчета времени выполнения (например, вычислений):

0 POKE 18,0:POKE 19,0:POKE 20,0

1 REM Здесь главная часть программы

32760 X=256\*256\*PEEK(18)+256\*PEEK(19)+PEEK(20)

32761 ?:??"-->Время вычислений ";X/50;" секунд"

65 – POKE 65,0 отключает звук, сопутствующий операциям чтения и записи при работе с магнитофоном или дисководом (нормально 00).

77 – POKE 77,129 включает "режим привлечения внимания" (автоматическое изменение цвета экрана); выключение производится величиной 0 или нажатием любой клавиши.

82 – POKE 82,x определяет левый край экрана (x может принимать значения от 0 до 39, нормальное значение – 2).

83 – POKE 83,x определяет правый край экрана (x может принимать значения от 0 до 39, нормальное значение – 39).

84 – в этой ячейке находится величина, соответствующая номеру строки экрана, в которой находится курсор (минимальное значение 0, максимальное

зависит от включенного графического режима).

85, 86 – содержат номер колонки экрана, в которой находится курсор.

87 – содержимое этой ячейки информирует о том, какой графический режим включен.

90 – позволяет указать строку, с которой начнется выполнение операторов DRAWTO и XIO 18 (заполнение фоном).

91, 92 – то же самое, но по отношению к колонке.

93 – содержит код символа, на котором находится курсор.

106 – объем свободной памяти в данный момент, исчисляется в блоках по 256 байт (страницах); можно использовать для выделения области памяти под данные или машинные программы – POKE 106, PEEK (106)-x, где x – количество страниц.

128, 129 – указатель конца памяти для Бейсика.

138, 139 – указатель таблицы переменных Бейсика, дает возможность заблокировать вывод текста программы на экран (необратимо!). Например:

1 GOTO 32760

2 REM Программа

32760 END

32761 FOR V=PEEK(130)+256\*PEEK(131) TO PEEK(132)+256\*PEEK(133):POKE V,155:NEXT V

32762 POKE PEEK(138)+256\*PEEK(139)+2,0

32763 SAVE "D:NAME":NEW или SAVE "C":NEW

Так заблокированная программа может быть запущена только инструкцией RUN "D:NAME" или RUN "C".

144, 145 – указатель начала памяти для Бейсика.

186, 187 – в этих ячейках хранится номер строки программы на Бейсике, на которой программа была прервана инструкциями STOP, TRAP, BREAK или в результате сбоя.

195 – если наступил сбой, в этой ячейке записывается номер ошибки. Использование ячеек 186, 187 и 195 позволяет написать элегантную процедуру диагностики ошибок:

1000 ERRORLINE = PEEK(186)+256\*PEEK(187)

1010 ERRORNUM = PEEK(195)

1020 ? "Ошибка номер ";ERRORNUM;" в строке ";ERRORLINE

Можно вместо сообщения о номере ошибки написать ее по-русски (о русских буквах в Atari мы писали в предыдущем выпуске "Компьютера". – Примеч. ред.).

200 – здесь хранится информация о цвете, используемом операторами PLOT и DRAWTO; вместо COLOR x можно применять POKE 200,x.

201 – POKE 201,x определяет величину табуляции в инструкциях PRINT с запятыми (не относится к клавише TAB), нормальное значение – 10, минимальное – 3:

10 PRINT 1,2,3,4,5

20 POKE 201,3

30 PRINT 1,2,3,4,5

251 – POKE 251,6 переключает вычисление тригонометрических функций на градусы, нормальное значение 0 – радианы.

Продолжение читайте в следующем выпуске.

Перевод Анджея Понлавского

# MULTISYSTEM

За переводные рубли  
пятой категории:

- Персональные  
компьютеры класса  
IBM XT, AT, 386
- Периферийное оборудова-  
ние для работы в системах  
XENIX/UNIX и оборудование  
для сети NOVELL
- Устройства для регенера-  
ции красящих лент принтера
- Множительная техника  
фирм CANON и RANK XEROX
- Телексы и телефаксы
- Теле-, аудио- и видео-  
аппаратура

Все это поставляет  
**MULTISYSTEM**

- Фирма обеспечивает  
гарантийное обслуживание  
поставляемого оборудования  
и послегарантийный сервис
- Наши специалисты  
производят монтаж  
компьютерных сетей
- Сотрудничество с  
**MULTISYSTEM** - гарантия  
Вашего успеха
- За более подробной тех-  
нической и коммерческой  
информацией просим  
обращаться непо-  
средственно  
в фирму



**MULTISYSTEM** -  
Przedsiębiorstwo pr-  
handlowe sp. z o.o.  
ul. Reymonta 12 a,  
01-842 Warszawa,  
tel. 35-77-83, tlx. 8

Представител  
СП "ТРАНСКОМ",  
103062 Москва,  
ул. Жуковского 7